

***Test of some numerical limiters for the conservative
PSM scheme for 4D Drift-Kinetic simulations.***

Jérôme Guterl — Jean-Philippe Braeunig

— Nicolas Crouseilles — Virginie Grandgirard — Guillaume Latu — Michel

Mehrenberger — Eric Sonnendrücker

N° 7467

November 2010

 **apport
de recherche**

Test of some numerical limiters for the conservative PSM scheme for 4D Drift-Kinetic simulations.

Jérôme Guterl^{*}, Jean-Philippe Braeunig^{†*} [‡]
, Nicolas Crouseilles^{†*}, Virginie Grandgirard[§], Guillaume Latu[§]
[†], Michel Mehrenberger^{*} [†], Eric Sonnendrücker^{*} [†]

Thème : Modélisation, analyse numérique
Équipe-Projet CALVI

Rapport de recherche n° 7467 — November 2010 — 63 pages

Abstract: The purpose of this work is simulation of magnetised plasmas in the ITER project framework. In this context, Vlasov-Poisson like models are used to simulate core turbulence in the tokamak in a toroidal geometry. This leads to heavy simulation because a 6D dimensional problem has to be solved, 3D in space and 3D in velocity. The model is reduced to a 5D gyrokinetic model, taking advantage of the particular motion of particles due to the presence of a strong magnetic field. However, accurate schemes, parallel algorithms need to be designed to bear these simulations. This paper describes a Hermite formulation of the conservative PSM scheme which is very generic and allows to implement different semi-Lagrangian schemes. We also test and propose numerical limiters which should improve the robustness of the simulations by diminishing spurious oscillations. We only consider here the 4D drift-kinetic model which is the backbone of the 5D gyrokinetic models and relevant to build a robust and accurate numerical method.

Key-words: numerical simulation, conservative scheme, ITER, plasma turbulence

^{*} IRMA, Université de Strasbourg, 7 rue René-Descartes, 67084 Strasbourg Cedex

[†] INRIA Nancy-Grand Est, 615 rue du Jardin Botanique, 54600 Villers-lès-Nancy

[‡] CEA/DIF Bruyères-le-Châtel, 91297 Arpajon Cedex

[§] CEA Cadarache, 13108 St Paul-lez-Durance Cedex

Test of some numerical limiters for the conservative PSM scheme for 4D Drift-Kinetic simulations.

Résumé : Ce travail concerne la simulation de plasmas magnétisés dans le cadre du projet ITER. Pour cette application, des modèles de type Vlasov-Poisson sont utilisés pour simuler la turbulence à coeur dans un tokamak, en géométrie toroidale. Ces études mènent à résoudre des problèmes dans un espace à 6 dimensions, 3D en espace 3D en vitesse, qui sont très lourds à simuler en terme de ressources informatiques. Le modèle est réduit à un modèle gyrocinétique 5D en exploitant les caractéristiques de ce plasma, dont le mouvement des particules est fortement influencé par la présence d'un champ magnétique intense. Cependant, il est nécessaire de mettre au point des schémas précis et des algorithmes parallèles pour mener ces simulations. Ce rapport décrit une formulation de type Hermite du schéma conservatif PSM qui est très générique et qui permet d'implémenter différents schémas semi-Lagrangiens. Nous testons et proposons également des limiteurs numériques de pente qui doivent accroître la robustesse des simulations en réduisant les oscillations d'origine numérique. Dans ce travail, nous l'utilisons pour résoudre le modèle drift-kinetic 4D, qui est le squelette du modèle gyrocinétique 5D. Ce modèle 4D est suffisamment pertinent pour la conception d'une méthode numérique robuste et précise pour le modèle 5D.

Mots-clés : simulation numérique, schéma conservatif, ITER, turbulence plasma

Contents

1	Introduction	4
2	Numerical schemes for the Vlasov equation	6
2.1	Directional splitting of the advection problem	6
2.2	Distribution function and phase space	7
2.3	Conservative semi-Lagrangian scheme principle	8
2.4	Finite volume form equivalence	9
2.5	Time scheme used in the Gysela code	9
3	The Hermite formalism applied to PSM and LAG	11
3.1	Computation of the flux $\phi_{i+1/2}$ with the Hermite formalism . . .	11
3.2	PSM scheme with the Hermite formalism	15
3.2.1	Hermite formalism using splines interpolation	15
3.2.2	Periodic boundary conditions	15
3.2.3	Non-Periodic boundary conditions: natural conditions . .	16
3.3	LAG scheme with the Hermite formalism	17
3.3.1	Hermite formalism using Lagrange interpolation	17
3.3.2	Boundary conditions	19
4	Limiters	21
4.1	ENTropic flux limiter (ENT)	21
4.2	UMEDA's limiter (UMEDA)	21
4.3	OScillation Limiter (OSL)	22
4.4	Slope Limited Splines (SLS)	23
5	Numerical results	25
5.1	Tools	25
5.2	Test of the limiters on constant 1D advections	26
5.3	4D Drift-kinetic model	34
5.4	Low resolution simulations	35
5.5	High resolution simulation	41
6	Conclusion	49
7	Annexes	50
7.1	Annexe A: Detailed LAG reconstruction	50
7.2	Positive displacement: $\beta > 0, \delta = 1, j_{k+1/2} = k$	54
7.3	Negative displacement $\delta = 0$	55
7.4	Annexe B: Algorithms	56
7.4.1	FluxComputation and FootCharacteristic	58
7.4.2	Hermite coefficient computation for natural conditions . .	58
7.4.3	Hermite coefficient computation for periodic conditions . .	59

1 Introduction

The ITER device is a tokamak designed to study controlled thermonuclear fusion. Roughly speaking, it is a toroidal vessel containing a magnetized plasma where fusion reactions occur. The plasma is kept out of the vessel walls by a magnetic field which lines have a specific helicoidal geometry. However, turbulence develops in the plasma and leads to thermal transport which decreases the confinement efficiency and thus needs a careful study. Plasma is constituted of ions and electrons, which motion is induced by the magnetic field. The characteristic mean free path is high, even compared with the vessel size, therefore a kinetic description of particles is required, see *Dimits* [6]. Then a full 6D Vlasov-Poisson model should be used for both ions and electrons to properly describe the plasma evolution. However, the plasma flow in presence of a strong magnetic field has characteristics that allow some physical assumptions to reduce the model. First, the Larmor radius, i.e. the radius of the cyclotronic motion of particles around magnetic field lines, can be considered as small compared with the tokamak size and the gyration frequency very fast compared to the plasma frequency. Thus this motion can be averaged (gyro-average) becoming the so-called guiding center motion. As a consequence, 6D Vlasov-Poisson model is reduced to a 5D gyrokinetic model by averaging equations in such a way the 6D toroidal coordinate system $(r, \theta, \phi, v_r, v_\theta, v_\phi)$ becomes a 5D coordinate system $(r, \theta, \phi, v_\parallel, \mu)$, with v_\parallel the parallel to the field lines component of the velocity and $\mu = m v_\perp^2 / 2B$ the adiabatic invariant which depends on the norm of the perpendicular to the field lines components of the velocity v_\perp^2 , on the magnetic field magnitude B and on the particles mass m . Moreover, the magnetic field is assumed to be steady and the mass of electrons m_e is very small compared to the mass of ions m_i . Thus the cyclotron frequency $\omega_{i,e} = q_{i,e} B / m_{i,e}$ is assumed to be much higher for electrons than for ions $\omega_e \gg \omega_i$. Therefore the electrons are assumed to be at Boltzmann equilibrium, i.e. the effect of the electrons cyclotronic motion is neglected. The 5D gyrokinetic model then reduces to a Vlasov like equation for ions guiding center motion:

$$\frac{\partial \bar{f}_\mu}{\partial t} + \frac{dX}{dt} \cdot \nabla_X \bar{f}_\mu + \frac{dv_\parallel}{dt} \partial_{v_\parallel} \bar{f}_\mu = 0 \quad (1)$$

where $\bar{f}_\mu(X, v_\parallel)$ is the ion distribution function with $X = (r, \theta, \phi)$, velocities dX/dt and dv_\parallel/dt define the guiding center trajectories.

If $\nabla_{(X, v_\parallel)} \cdot (dX/dt, dv_\parallel/dt)^t = 0$, then the model is termed as conservative.

This equation for ions is coupled with a quasi-neutrality equation for the electric potential $\Phi(R)$ on real particles position, with $R = X - \rho_L$ (with ρ_L the Larmor radius) :

$$-\frac{1}{B\omega_i} \nabla_\perp \cdot (n_0 \nabla_\perp \Phi) + \frac{e}{\kappa T_e} (\Phi - \langle \Phi \rangle_\phi) = \int \bar{f}_\mu d\mu dv_\parallel - n_0 \quad (2)$$

where n_0 is the equilibrium electronic density, T_e the electronic temperature, e the electronic charge, κ the Boltzmann constant for electrons and ω_i the cyclotronic frequency for ions.

These equations are of a simple form, but they have to be solved very efficiently because of the 5D space and the large characteristic time scales considered. However, the adiabatic invariant μ acts as a parameter, thus it could

easily be parallelized. Moreover, we can see that for each μ , we have to solve a 4D advection equation, as accurately as possible but also taking special care on mass and energy conservation, especially in this context of large characteristic time scales. The maximum principle that exists at the continuous level for the Vlasov equation should also be carefully studied at discrete level because there is no physical dissipation process in this model that might dissipate over/undershoots of the scheme. These studies will be achieved first on a relevant reduced model, the 4D drift-kinetic model which corresponds to (1) with $\mu = 0$ instead a range of μ values (theoretically \mathbb{R}^+). This work follows those of *Grandgirard et al* in the GYSELA code, see [8] and [9]. The geometrical assumptions of this model for ion plasma turbulence are a cylindrical geometry with coordinates $(r, \theta, z, v_{\parallel})$ and a constant magnetic field $B = B_z e_z$, where e_z is the unit vector in z direction. In this collisionless plasma, the trajectories are governed by the guiding center (GC) trajectories:

$$\frac{dr}{dt} = v_{GC_r}; \quad r \frac{d\theta}{dt} = v_{GC_\theta}; \quad \frac{dz}{dt} = v_{\parallel}; \quad \frac{dv_{\parallel}}{dt} = \frac{q_i}{m_i} E_z \quad (3)$$

with $v_{GC} = (E \times B)/B^2$ and $E = -\nabla\Phi$ with Φ the electric potential. The Vlasov equation governing this system, with the ion distribution function $f(r, \theta, z, v_{\parallel}, t)$, is the following:

$$\partial_t f + v_{GC_r} \partial_r f + v_{GC_\theta} \partial_\theta f + v_{\parallel} \partial_z f + \frac{q_i}{m_i} E_z \partial_{v_{\parallel}} f = 0. \quad (4)$$

This equation is coupled with a quasi-neutrality equation for the electric potential $\Phi(r, \theta, z)$ that reads the same as for the 5D gyrokinetic model (2) with $\mu = 0$.

Let us notice that the 4D velocity field $a = (v_{GC_r}, v_{GC_\theta}, v_{\parallel}, q/m_i E_z)^t$ is divergence free:

$$\nabla \cdot a = \frac{1}{r} \partial_r (r v_{GC_r}) + \frac{1}{r} \partial_\theta (v_{GC_\theta}) + \partial_z v_{\parallel} + \partial_{v_{\parallel}} (q_i/m_i E_z) = 0 \quad (5)$$

because of variable independence $\partial_{v_{\parallel}} E_z = \partial_{v_{\parallel}} (\partial_z \Phi(r, \theta, z)) = 0$ and $\partial_z v_{\parallel} = 0$. Moreover we have $v_{GC} = (E \times B)/B^2$, with $E = -\nabla\Phi$ and $B = B_z e_z$, thus:

$$v_{GC_r} = \frac{1}{B_z} \left(-\frac{1}{r} \partial_\theta \Phi \right) \quad \text{and} \quad v_{GC_\theta} = \frac{1}{B_z} (\partial_r \Phi) \quad (6)$$

and

$$\nabla_{r\theta} \cdot a = \frac{1}{r} \partial_r (r v_{GC_r}) + \frac{1}{r} \partial_\theta (v_{GC_\theta}) = \frac{1}{r B_z} (\partial_r (r (-1/r) \partial_\theta \Phi) + \partial_\theta (\partial_r \Phi)) = 0. \quad (7)$$

Therefore, one can write an equivalent conservative equation to the preceding Vlasov equation (4):

$$\partial_t f + \partial_r (v_{GC_r} f) + \partial_\theta (v_{GC_\theta} f) + \partial_z (v_{\parallel} f) + \partial_{v_{\parallel}} \left(\frac{q_i}{m_i} E_z f \right) = 0. \quad (8)$$

This conservative system will be discretized using a conservative semi - Lagrangian scheme. Following [1] and [16], we consider two conservative schemes, which are fourth order in space:

- **LAG**: LAGrange polynom method, which uses Lagrangian polynoms to reconstruct the distribution function.
- **PSM**: Parabolic Splines Method, which uses cubic splines to reconstruct the distribution function.

These schemes are designed to solve conservative models and they allow a directional splitting of the Vlasov equation (8). This equation will be solved by using D (dimensions of space) 1D conservative steps, discretized by using the 1D schemes (LAG or PSM). At the continuous level, each 1D step has no maximum principle, it is only the solution after all D directional steps, the solution of the Vlasov equation, that should satisfy a maximum principle [1]. However, high order schemes may create spurious oscillations leading to break this maximum principle. A flux limiting procedure may improve the discrete solution, which may be closer to the maximum principle in the sense of showing less spurious oscillations. The limiter does not ensure a maximum principle in 1D, but should decrease the oscillations amplitude created by the scheme by locally adding numerical diffusion. That leads us to investigate and compare in details the properties of many limiters, which depend mainly on two issues:

- How to make the high order schemes degenerate into a more diffusive scheme?
- How to detect in the function profile the location where the scheme will produce oscillations?

We first describe a Hermite formalism proposed by [4] applied to the PSM and LAG schemes leading to a finite volume form equivalent to the original semi-Lagrangian schemes. This formalism is very efficient to introduce limiters in the PSM or LAG schemes. Consequently, we will compare some limiters, focusing in particular on the OSillations limiter (OSL) proposed by [4]. We also propose and investigate a new limiter (Slope Limited Spline, SLS), based on classical slope limiting methods. We evaluate the performances of each limiter using a benchmark developed in the Gysela code, which runs a 4D drift-kinetic model [9].

The outline of this paper is the following : in section 2 will be recalled some important properties concerning the conservative form of the Vlasov equations. Then the Hermite formalism applied to LAG and PSM schemes will be explored. In section 4, some limiters are described and they are further investigated in section 5 in the context of a 4D drift-kinetic model. At last we will comment on numerical results.

2 Numerical schemes for the Vlasov equation

2.1 Directional splitting of the advection problem

In a phase space of dimension D , we consider a distribution f which is advected by a velocity field \mathbf{a} . The model taken into account satisfies $\nabla \cdot \mathbf{a} = 0$.

$$t \in \mathbb{R}^+, \mathbf{x} \in \mathbb{R}^D, \mathbf{a}(\mathbf{x}, t) \in \mathbb{R}^D \begin{cases} \partial_t f + \nabla_x \cdot (\mathbf{a}f) = 0 \\ \nabla \cdot \mathbf{a} = 0 \\ f(x, t) \geq 0 \end{cases} \quad (9)$$

For instance in cylindrical geometry, $\mathbf{x} = (r, \theta, z, v_{\parallel})$ considering the 4D problem dealt by the Gysela code. In the next sections, we use a directional splitting following [1] by solving the conservative system (9) by D separate 1D problems for each phase space direction which are still under a conservative form. So formally, we will consider the problem (10) for each of the D directions. The generic direction is named x .

$$t \in \mathbb{R}^+, x \in \mathbb{R}, a(x, t) \in \mathbb{R} \begin{cases} \partial_t f + \frac{\partial(a(x, t)f(x, t))}{\partial x} = 0 \\ f(x, t) \geq 0 \end{cases} \quad (10)$$

In this context, we don't have in general $\forall k \in \llbracket 1, D \rrbracket, \frac{\partial a(x, t)}{\partial x_k} = 0$, but only $\nabla \cdot \mathbf{a} = 0$.

2.2 Distribution function and phase space

We divide one direction of the phase space, generically x with a constant step Δx to get a regular mesh. The cells are numbered by an integer i from 0 to N and the cell faces by an one-half integer $i \pm 1/2$ (see fig. 1). Hence we have $N+1$ cells and $N+2$ faces.

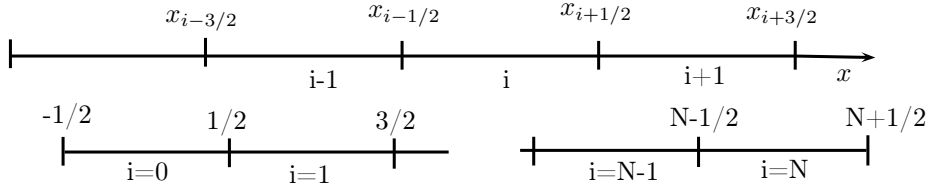


Figure 1: Mesh grid on the x phase space direction (top) and beginning (left bottom) and end (right bottom) of the mesh grid.

We note the distribution function at the time t and at the position x in the phase space: $\mathbf{g}(\mathbf{t}, \mathbf{x})$. We discretize the time space with a constant time step Δt . Writing $t^n = n\Delta t$, we then note $g^n(x) = g(t^n, x)$ the value of the distribution function at t^n . Using the previous discretization, we define the distribution function at the cells faces as $g_{i+1/2}^n = g^n(x_{i+1/2})$ (fig. 2.2). At last, we define the average of the distribution function on one cell i at t^n by:

$$\bar{g}_i^n = \frac{1}{\Delta x} \int_{x_{i-1/2}}^{x_{i+1/2}} g^n(x) dx, \quad i = 0 \dots N$$

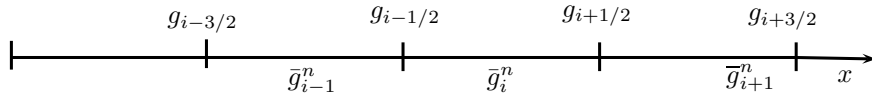


Figure 2: Normative example for the distribution g in the cells and at the nodes

2.3 Conservative semi-Lagrangian scheme principle

The mass conservation in a lagrangian volume of the phase space between t^n and t^{n+1} reads as follows:

$$\int_{Vol^{n+1}} g(x, t^{n+1}) dx = \int_{Vol^n} g(x, t^n) dx \quad (11)$$

$$\text{with } Vol^n = \{X(x^{n+1}, t^n) | X(x^{n+1}, t^{n+1}) \in Vol^{n+1}\}$$

where $X(x, t^{n+1})$ describes the characteristic curve which passes by $x^{n+1} = X(x^{n+1}, t^{n+1})$ at t^{n+1} . Thus $X(x^{n+1}, t^n)$ is the point by which the trajectory passes at t^n such as this trajectory also passes by x^{n+1} at t^{n+1} . The characteristic curves are obtained by solving the following equation:

$$\frac{dX(x, t)}{dt} = a(x, t)$$

with a given initial condition. We consider hereafter the trajectories which pass by the cells faces $x_{i+1/2}$ at t^{n+1} which are noted: $x_{i+1/2} = X(x_{i+1/2}^{n+1}, t^{n+1})$ and we introduce $x_{i+1/2}^*$ which is the point on each characteristic curve at the time t^n :

$$x_{i+1/2}^* = X(x_{i+1/2}^{n+1}, t^n).$$

The conservation equation (11) can thus be written using a 1D discretized form:

$$\Delta x \bar{g}_i^{n+1} = \int_{x_{i-1/2}}^{x_{i+1/2}} g^{n+1}(y) dy = \int_{x_{i-1/2}^*}^{x_{i+1/2}^*} g^n(y) dy \quad (12)$$

where

$$\begin{cases} x_{i+1/2}^* - x_{i-1/2}^* = Vol^n \\ x_{i+1/2} - x_{i-1/2} = Vol^{n+1} = \Delta x \end{cases}$$

In the conservative semi-Lagrangian formalism, the right hand side of equation (12) is numerically computed as follows:

$$\Delta x \bar{g}_i^{n+1} = \int_{x_{i-1/2}^*}^{x_{i+1/2}^*} g^n(y) dy = G(x_{i+1/2}^*) - G(x_{i-1/2}^*) \quad (13)$$

where $G(x)$ is the cumulative or primitive function of g defined as:

$$G(x) = \int_{x_{-1/2}}^x g(y) dy.$$

This primitive function can be computed exactly at each cell face of the mesh:

$$G(x_{i+1/2}) = G(x_{-1/2}) + \sum_{k=0}^i \Delta x \bar{g}_k^n.$$

These values at faces are then interpolated by basis functions to obtain an approximate reconstruction $G_h(x)$ of $G(x)$ for any x .

For instance, the PSM scheme uses cubic splines and the LAG scheme uses third order lagrangian polynoma as interpolation functions to obtain the reconstructed function $G_h(x)$.

2.4 Finite volume form equivalence

The equation (12) can be split in three terms:

$$\Delta x \bar{g}_i^{n+1} = \underbrace{\int_{x_{i-1/2}^*}^{x_{i-1/2}} g^n(y) dy}_{\phi_{i-1/2}} + \underbrace{\int_{x_{i-1/2}}^{x_{i+1/2}} g^n(y) dy}_{\bar{g}_i^n \Delta x} + \underbrace{\int_{x_{i+1/2}}^{x_{i+1/2}^*} g^n(y) dy}_{-\phi_{i+1/2}} \quad (14)$$

We name $\phi_{i+1/2}$ the following quantity:

$$\phi_{i+1/2} = \int_{x_{i+1/2}^*}^{x_{i+1/2}} g^n(y) dy.$$

We call it 'flux' since it represents the algebraic quantity which is carried through the node $x_{i+1/2}$, by identification with the finite volume formalism.

The equation (14) is a finite volume equation (see fig. 3) meaning that the new value of the distribution function \bar{g}_i^{n+1} in the cell i is the sum of its value at time t^n and the incoming or outgoing flux $\phi_{i\pm 1/2}$.

$$\bar{g}_i^{n+1} = \bar{g}_i^n - \left(\frac{\phi_{i+1/2} - \phi_{i-1/2}}{\Delta x} \right). \quad (15)$$

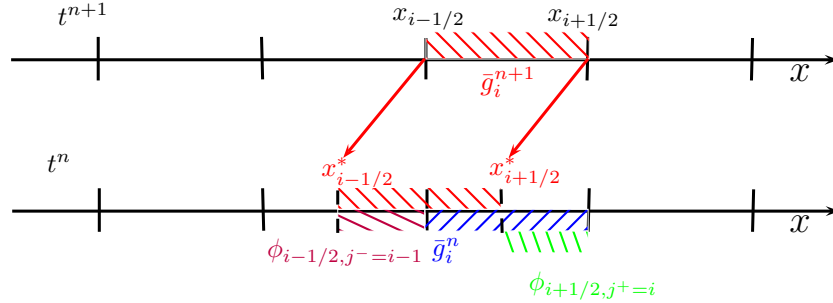


Figure 3: Conservative evolution of a finite volume scheme

As for the semi-Lagrangian formalism (13), the integrals to be computed to obtain the fluxes can be approximated by a reconstruction of the primitive of the distribution function G_h :

$$\phi_{i+1/2} = \int_{x_{i+1/2}^*}^{x_{i+1/2}} g^n(y) dy \approx G_h(x_{i+1/2}) - G_h(x_{i+1/2}^*). \quad (16)$$

2.5 Time scheme used in the Gysela code

The time scheme has been modified, because the classical second order leap-frog algorithm used in the Gysela code is not robust enough to deal with the conservative semi-Lagrangian schemes considered here:

$$\bar{g}_i^{n+1} = \bar{g}_i^{n-1} - \left(\frac{\phi_{i+1/2}^n - \phi_{i-1/2}^n}{\Delta x} \right)$$

Moreover, this time scheme enforces a constant time step because it involves three different time steps, which is quite restrictive for high iteration numbers simulations. We thus turn the time scheme to a Predictor-Corrector (or Runge Kutta order 2) method, which allows to use a variable time step Δt^n computed at time t^n . It is computed according to a CFL like condition necessary for the finite volume scheme stability which assesses that the maximum displacement in the domain is less than a fraction of the cells size (using a regular mesh):

$$\Delta t^n = CFL \min_{d=1,D} \left(\frac{\Delta x_d}{\max_x(a_d^n(x))} \right) \quad (17)$$

with Δx_d the space step in direction d and a_d^n the velocity at time t^n in the space direction $d \in [1, D]$. CFL is a coefficient which $0 < CFL < 1$.

Remark 1. *The finite volumes scheme form (15) and the semi-Lagrangian scheme (13) are strictly equivalent, since the displacement is restricted to $CFL \leq 1$. The finite volumes scheme (15) is not defined for displacements bigger than one cell, with $CFL > 1$, although the semi-Lagrangian scheme (13) could be written for any time step. However, the stability of both schemes with $CFL > 1$ in a general situation is not demonstrated.*

Predictor-Corrector Algorithm:

- At beginning of the iteration at t^n , we compute Δt^n according to (17).
- Prediction step : we compute a order 1 in time approximation of the solution at time $t^{n+1/2}$ with half a time step according to values \bar{g}_i^n and cell faces fluxes $\phi_{d,i+1/2}^n$ in all direction $d \in [1, D]$ at time t^n :

$$\bar{g}_i^{n+1/2} = \bar{g}_i^n - \sum_{d=1}^D \left(\frac{\phi_{d,i+1/2}^n - \phi_{d,i-1/2}^n}{\Delta x_d} \right)$$

We compute the electric potential at same time $t^{n+1/2}$:

$$-\frac{1}{B\omega_i} \nabla_{\perp} \cdot (n_0 \nabla_{\perp} \Phi^{n+1/2}) + \frac{e}{\kappa T_e} (\Phi^{n+1/2} - \langle \Phi^{n+1/2} \rangle_{\phi}) = \int \bar{g}^{n+1/2} d\mu dv_{\parallel} - n_0$$

- Correction step : we compute an order 2 in time approximation of the solution at time t^{n+1} according to values \bar{g}_i^n at time t^n and cell faces fluxes $\phi_{d,i+1/2}^{n+1/2}$ in all direction $d \in [1, D]$ at time $t^{n+1/2}$:

$$\bar{g}_i^{n+1} = \bar{g}_i^n - \sum_{d=1}^D \left(\frac{\phi_{d,i+1/2}^{n+1/2} - \phi_{d,i-1/2}^{n+1/2}}{\Delta x_d} \right)$$

We compute the electric potential at same time t^{n+1} :

$$-\frac{1}{B\omega_i} \nabla_{\perp} \cdot (n_0 \nabla_{\perp} \Phi^{n+1}) + \frac{e}{\kappa T_e} (\Phi^{n+1} - \langle \Phi^{n+1} \rangle_{\phi}) = \int \bar{g}^{n+1} d\mu dv_{\parallel} - n_0$$

3 The Hermite formalism applied to PSM and LAG

Two schemes can be used to compute the new distribution function (reconstructing the flux or the primitive function):

- The Parabolic Splines Method called PSM, which uses cubic splines functions for the interpolations.
- The Lagrangian method (LAG), which uses third order Lagrange polynoms for the interpolations.

The PSM scheme [1] or the LAG scheme [4] are conservative semi-Lagrangian schemes that only differs by the interpolation functions used for the reconstruction step. We aim to study some limiters for both schemes and using the Hermite formalism. The Hermite formalism is a generic formulation for the interpolation polynoms. Indeed, the scheme LAG or PSM in this formalism are set only by the way of computing the distribution function at the faces $g_{i+1/2}$. Thus we can easily use them simultaneously in a code.

We rewrite hereafter this generic formulation, based on the Hermite formalism for the conservative schemes PSM and LAG, proposed in [4]. First, we give the expression of the flux $\phi_{i+1/2}$, then we propose an application to the PSM and LAG schemes.

3.1 Computation of the flux $\phi_{i+1/2}$ with the Hermite formalism

Following [16] and [4], we reconstruct the distribution function g with a second order polynomial P_k , which interpolates the distribution function in the cell k . We first assume that the reconstructed function is continuous at the cell faces. We name the value of g at the cell faces $g_{k+1/2}$. At the end of the section, we will give the general formula for a discontinuous function at the cell faces. The distribution function is approximated by a second order polynomial in cell k , which corresponds to interpolate the primitive of the distribution function with a third order polynomial:

$$\forall k, P_k(z) = c^{(k)} + b^{(k)}z + a^{(k)}z^2 \text{ with } z \in [0, \Delta x]$$

The distribution function g is continuous at the faces. That implies:

$$\begin{cases} P_k(0) = g_{k-1/2} \\ P_k(\Delta x) = g_{k+1/2} \end{cases}$$

A third condition on the polynom comes from the mass conservation in the cell k , that reads:

$$\frac{1}{\Delta x} \int_{x_{k-1/2}}^{x_{k+1/2}} P_k(y) dy = \bar{g}_k$$

Therefore, the polynomial coefficients can be written:

$$\begin{aligned} a^{(k)} &= (3g_{k-1/2} + 3g_{k+1/2} - 6\bar{g}_k) / \Delta x^2 \\ b^{(k)} &= (-4g_{k-1/2} - 2g_{k+1/2} + 6\bar{g}_k) / \Delta x \\ c^{(k)} &= g_{k-1/2} \end{aligned} \tag{18}$$

Finally, $\forall x \in [x_{k-1/2}, x_{k+1/2}]$, $P_k(x)$ is an approximation of the distribution function g in the cell k . This approximation depends on $g_{k-1/2}$, $g_{k+1/2}$, and \bar{g}_k . Using this interpolation, we are able to evaluate the flux $\phi_{k+1/2}$:

$$\phi_{k+1/2} = \int_{x_{k+1/2}^*}^{x_{k+1/2}} g^n(y) dy. \quad (19)$$

We take the additional assumption that the displacement at the cell face $k + 1/2$, i.e. $\alpha_{k+1/2} = x_{k+1/2} - x_{k+1/2}^*$ satisfies:

$$|\alpha_{k+1/2}| \leq \Delta x$$

It is necessary to know in which cell j ($j = k$ or $j = k + 1$ since the displacement $|\alpha_{k+1/2}| \leq \Delta x$) is located the foot $x_{k+1/2}^*$ of the characteristic that passes by $x_{k+1/2}$ at time t^{n+1} . We can thus give an approximation of the fluxes (19) towards the face $j + 1/2$ based on the set of polynoms P_k :

$$\begin{aligned} \exists j \mid x_{k+1/2}^* &\in [x_{j-1/2}, x_{j+1/2}] \\ \phi_{k+1/2} &= \int_{x_{k+1/2}^* - x_{j-1/2}}^{x_{k+1/2} - x_{j-1/2}} g^n(Y) dY \approx \int_{x_{k+1/2}^* - x_{j-1/2}}^{x_{k+1/2} - x_{j-1/2}} P_j(Y) dY \end{aligned}$$

with the change of variable $Y = y - x_{j-1/2} \in [0, \Delta x]$.

We exhibit hereafter the flux $\phi_{k+1/2}$ function of the $g_{k-1/2}$, $g_{k+1/2}$, and \bar{g}_k . We note $j_{k+1/2}$ the cell where is located the foot of the characteristic passing by the face $k + 1/2$:

$$x_{k+1/2}^* \in [x_{j_{k+1/2}-1/2}, x_{j_{k+1/2}+1/2}]$$

The index $j_{k+1/2}$ indicates from which cell the flow is coming. For instance, a negative displacement $\alpha_{i+1/2} < 0$ on the face $k + 1/2$ means that the flow comes from the cell $k + 1$, so that $j_{k+1/2} = k + 1$.

So we have:

- If $x_{k+1/2}^* < x_{k+1/2}$ then
 - $j_{k+1/2} = k$
 - $\phi_{k+1/2} = \int_{x_{k+1/2}^* - x_{k+1/2}}^{x_{k+1/2} - x_{k-1/2} = \Delta x} P_k(Y) dY$
- If $x_{k+1/2}^* > x_{k+1/2}$ then
 - $j_{k+1/2} = k + 1$
 - $\phi_{k+1/2} = \int_{x_{k+1/2}^* - x_{k+1/2}}^{x_{k+1/2} - x_{k+1/2} = 0} P_{k+1}(Y) dY$

We introduce

$$\delta = \frac{x_{k+1/2} - x_{j_{k+1/2}-1/2}}{\Delta x},$$

thus $\delta = 0$ or $\delta = 1$ indicates the upwinding direction.

By introducing a normalized displacement

$$\beta = \frac{x_{k+1/2}^* - x_{k+1/2}}{\Delta x},$$

we can write the flux:

$$\phi_{k+1/2}(\beta) = \int_{(\delta-\beta)\Delta x}^{\delta\Delta x} P_{j_{k+1/2}}(Y) dY$$

That leads to:

$$\frac{\phi_{k+1/2}}{\Delta x} = a_{j_{k+1/2}}\beta + b_{j_{k+1/2}}(-\beta^2 + 2\beta\delta) + c_{j_{k+1/2}}(\beta^3 + 3\delta^2\beta - 3\beta^2\delta) \quad (20)$$

with

$$\begin{cases} a_{j_{k+1/2}} = g_{j_{k+1/2}-1/2} \\ b_{j_{k+1/2}} = \frac{-4g_{j_{k+1/2}-1/2} + 2g_{j_{k+1/2}+1/2} + 6\bar{g}_{j_{k+1/2}}}{2} \\ c_{j_{k+1/2}} = \frac{3g_{j_{k+1/2}-1/2} + 3g_{j_{k+1/2}+1/2} - 6\bar{g}_{j_{k+1/2}}}{3} \end{cases}$$

By ordering differently the polynomial expression (20) and replacing the coefficients by their values, we try to get an equivalent formulation to the flux expressions proposed by [4].

$$\begin{aligned} \phi_{k+1/2,j_{k+1/2}}(\beta) &= \Delta x [g_{j_{k+1/2}-1/2}(\beta(1-\delta) + \beta^2(2-3\delta) + \beta^3) \\ &\quad + g_{j_{k+1/2}+1/2}(\beta\delta + \beta^2(1-3\delta) + \beta^3) \\ &\quad + \bar{g}_{j_{k+1/2}}(\beta^2(-3+6\delta) + \beta^3(-2))] \end{aligned}$$

with

$$\begin{aligned} \beta &= \frac{x_{k+1/2} - x_{k+1/2}^*}{\Delta x} \in [-1, 1] \\ \delta &= \begin{cases} 0 & \text{if } x_{k+1/2} < x_{k+1/2}^* \\ 1 & \text{if } x_{k+1/2} > x_{k+1/2}^* \end{cases} \end{aligned}$$

It's obvious that β depends on k hence formally $\beta = \beta_{k+1/2}$

Positive displacement $\alpha > 0$ i.e. $\delta = 1$

We have $j_{k+1/2} = k$, hence

$$\begin{aligned} \frac{\phi_{k+1/2}(\beta)}{\Delta x} &= g_{k-1/2}(\beta^2(\beta-1)) \\ &\quad + g_{k+1/2}(\beta(1-\beta)^2) \\ &\quad + \bar{g}_k(\beta^2(3-2\beta)) \end{aligned}$$

Negative displacement $\alpha < 0$ i.e. $\delta = 0$

We have $j_{k+1/2} = k+1$, hence

$$\begin{aligned} \frac{\phi_{k+1/2}(\beta)}{\Delta x} &= g_{k+1/2}(\beta(\beta+1)^2) \\ &\quad + g_{k+3/2}(\beta^2(1+\beta)) \\ &\quad + \bar{g}_{k+3/2}(\beta^2(-3-2\beta)) \end{aligned}$$

Thus, we have established a generic formulation for the flux considering displacements smaller than one cell. We have assumed that the distribution function is continuous at the cell face. This generic expression is equivalent to the formulation proposed by [4].

Hermite formalism for a discontinuous reconstruction

Performing the same calculations, we can extend the previous formalism to a distribution function which is not continuous at the faces. We name $g_{k+1/2}^+$ and $g_{k+1/2}^-$ respectively the left and right values of the distribution function at the cell face $k + 1/2$ (see fig. 4). In general that means:

$$g_{k+1/2}^+ \neq g_{k+1/2}^-$$

The conditions on the polynomials at the cell faces are thus changed to:

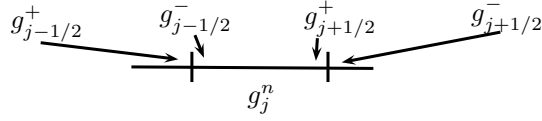


Figure 4: Asymmetric node

$$\begin{cases} P_k(0) = g_{k-1/2}^- \\ P_k(\Delta x) = g_{k+1/2}^+ \end{cases}$$

That leads to the final formulation:

$$\begin{aligned} \phi_{k+1/2, j_{k+1/2}}(\beta) &= \Delta x [g_{j_{k+1/2}-1/2}^- (\beta(1-\delta) + \beta^2(2-3\delta) + \beta^3) \\ &\quad + g_{j_{k+1/2}+1/2}^+ (\beta\delta + \beta^2(1-3\delta) + \beta^3) \\ &\quad + \bar{g}_{j_{k+1/2}} (\beta^2(-3+6\delta) + \beta^3(-2))] \end{aligned} \quad (21)$$

where

$$\begin{aligned} \beta &= \frac{\alpha_{i+1/2}}{\Delta x} = \frac{x_{k+1/2} - x_{k+1/2}^*}{\Delta x} \in [-1, 1] \\ \delta &= \begin{cases} 0 & \text{if } x_{k+1/2} < x_{k+1/2}^* \\ 1 & \text{if } x_{k+1/2} > x_{k+1/2}^* \end{cases} \end{aligned}$$

Positive displacement $\alpha > 0$ i.e. $\delta = 1$

We have $j_{k+1/2} = k$, hence

$$\begin{aligned} \frac{\phi_{k+1/2}(\beta)}{\Delta x} &= g_{k-1/2}^- (\beta^2(\beta-1)) \\ &\quad + g_{k+1/2}^+ (\beta(1-\beta)^2) \\ &\quad + \bar{g}_k (\beta^2(3-2\beta)) \end{aligned}$$

Negative displacement $\alpha < 0$ i.e. $\delta = 0$

We have $j_{k+1/2} = k+1$, hence

$$\begin{aligned} \frac{\phi_{k+1/2}(\beta)}{\Delta x} &= g_{k+1/2}^- (\beta(\beta+1)^2) \\ &\quad + g_{k+3/2}^+ (\beta^2(1+\beta)) \\ &\quad + \bar{g}_{k+1} (\beta^2(-3-2\beta)) \end{aligned}$$

3.2 PSM scheme with the Hermite formalism

3.2.1 Hermite formalism using splines interpolation

The type of scheme expressed in the hermite formalism (PSM or LAG) only depends on the manner the face values $g_{k+1/2}^+$ and $g_{k+1/2}^-$ are computed. Here the PSM scheme has the property that the reconstructed function has continuous derivatives of the distribution function $g(x, t)$ at cell faces [16]. Remembering that the distribution function g is reconstructed by a second order polynomial $g(x) \approx P_i(x)$ in the cell i , the continuity of the derivative at face $i + 1/2$ can be written:

$$\frac{dP_i(Y)}{dY} \Big|_{Y=\Delta x} = \frac{dP_{i+1}(Y)}{dY} \Big|_{Y=0}$$

Using the polynomial coefficients expression (18) we obtain:

$$\begin{aligned} \frac{dP_i(Y)}{dY} \Big|_{Y=\Delta x} &= \frac{dP_{i+1}(Y)}{dY} \Big|_{Y=0} \\ \Leftrightarrow 2a_i\Delta x + b_i &= b_{i+1} \\ \Leftrightarrow g_{i-1/2} + 4g_{i+1/2} + g_{i+3/2} &= 3(\bar{g}_i + \bar{g}_{i+1}) \end{aligned}$$

This PSM formulation regardless of the boundary conditions is equivalent to the semi-Lagrangian PSM formalism, used in [1] for instance. A rigorous proof of the equivalence between the two formulations is furnished by [4], except for the boundary conditions.

3.2.2 Periodic boundary conditions

In this section, we present the way of imposing the boundary conditions. Extending boundary conditions for the PSM scheme to the Hermite formalism is not simple, especially to get a complete equivalence between the Hermite formalism and the semi-Lagrangian formalism (13) [1].

We named G the primitive function which is defined as:

$$G(x) = \int_{x_{-1/2}}^x g(x)dx \quad (22)$$

We also define the mesh fitted to a periodic domain(fig. 5):

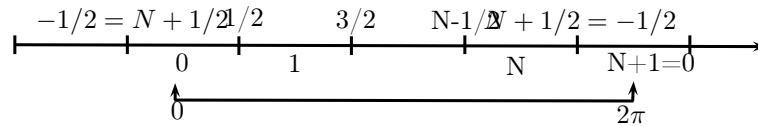


Figure 5: Mesh for a periodic domain

The periodic boundary conditions for the semi-Lagrangian scheme are approximated by conditions on the primitive function derivatives:

$$\begin{cases} G'(x_{-1/2}) = G'(x_{N+1/2}) \\ G''(x_{-1/2}) = G''(x_{N+1/2}) \end{cases}$$

which is equivalent to set continuity of the distribution function and its first derivative:

$$\begin{cases} g(x_{-1/2}) = g(x_{N+1/2}) \\ g'(x_{-1/2}) = g'(x_{N+1/2}) \end{cases}$$

The periodic boundary conditions for the Hermite formalism are then obtained by setting the same constraint on the polynomial reconstruction:

$$\begin{aligned} & \begin{cases} P_0(0) = P_N(\Delta x) \\ \frac{dP_0(Y)}{dY}|_{Y=0} = \frac{dP_N(Y)}{dY}|_{Y=\Delta x} \end{cases} \\ \Leftrightarrow & \begin{cases} g_{-1/2} = g_{N+1/2} \\ -4g_{-1/2} - 2g_{1/2} + 6\bar{g}_0 = 2g_{N-1/2} + 4g_{N+1/2} - 6\bar{g}_N \end{cases} \end{aligned}$$

The complete linear system to solve to obtain values at nodes is then the following

$$\begin{aligned} & \begin{cases} g_{i-1/2} + 4g_{i+1/2} + g_{i+3/2} = 3(\bar{g}_i + \bar{g}_{i+1}), i = 0 \cdots N-1 \\ g_{-1/2} = g_{N+1/2} \\ -4g_{-1/2} - 2g_{1/2} + 6\bar{g}_0 = 2g_{N-1/2} + 4g_{N+1/2} - 6\bar{g}_N \end{cases} \\ \Rightarrow & \begin{cases} g_{i-1/2} + 4g_{i+1/2} + g_{i+3/2} = 3(\bar{g}_i + \bar{g}_{i+1}), i = 0 \cdots N-2 \\ g_{N-3/2} + 4g_{N-1/2} + g_{-1/2} = 3(\bar{g}_{N-1} + \bar{g}_N) \\ g_{N-1/2} + 4g_{-1/2} + g_{1/2} = 3(\bar{g}_0 + \bar{g}_1) \end{cases} \end{aligned}$$

The matricial system $[A]X = B$ of dimension $N+1$ is finally:

$$\underbrace{\begin{pmatrix} 4 & 1 & 0 & 0 & \cdots & 1 \\ 1 & 4 & 1 & 0 & \cdots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \cdots & 0 & 1 & 4 & 1 \\ 1 & \cdots & 0 & 0 & 1 & 4 \end{pmatrix}}_{[A]} \underbrace{\begin{pmatrix} g_{-1/2} \\ g_{1/2} \\ \vdots \\ \vdots \\ g_{N-3/2} \\ g_{N-1/2} \end{pmatrix}}_X = \underbrace{\begin{pmatrix} 3(\bar{g}_N + \bar{g}_0) \\ 3(\bar{g}_0 + \bar{g}_1) \\ \vdots \\ \vdots \\ 3(\bar{g}_{N-2} + \bar{g}_{N-1}) \\ 3(\bar{g}_{N-1} + \bar{g}_N) \end{pmatrix}}_B$$

3.2.3 Non-Periodic boundary conditions: natural conditions

The natural boundary condition for the PSM scheme is approximated by vanishing the second derivative of the primitive function, what corresponds to annulate the first derivative of the distribution function g , at the boundaries:

$$\begin{cases} G'''(x_{-1/2}) = 0 \\ G'''(x_{N+1/2}) = 0 \end{cases}$$

The corresponding natural boundary conditions for the Hermite formalism are:

$$\begin{cases} \frac{dP_0(Y)}{dY}|_{Y=0} = 0 \\ \frac{dP_N(Y)}{dY}|_{Y=\Delta x} = 0 \end{cases} \Leftrightarrow \begin{cases} -4g_{-1/2} - 2g_{1/2} + 6\bar{g}_0 = 0 \\ 2g_{N-1/2} + 4g_{N+1/2} - 6\bar{g}_N = 0 \end{cases}$$

The system to solve to obtain the values of the function at the nodes is then:

$$\begin{cases} g_{i-1/2} + 4g_{i+1/2} + g_{i+3/2} = 3(\bar{g}_i + \bar{g}_{i+1}), i = 0 \dots N-1 \\ 4g_{-1/2} + 2g_{1/2} = 6\bar{g}_0 \\ 2g_{N-1/2} + 4g_{N+1/2} = 6\bar{g}_N \end{cases}$$

Considering $g_{-1/2} = \bar{g}_0$ and $g_{N+1/2} = \bar{g}_N$, we solve from the following linear system of dimension $N+2$:

$$\underbrace{\begin{pmatrix} 4 & 2 & 0 & 0 & \dots & 0 \\ 1 & 4 & 1 & 0 & \dots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \dots & 0 & 1 & 4 & 1 \\ 0 & \dots & 0 & 0 & 2 & 4 \end{pmatrix}}_{[A]} \underbrace{\begin{pmatrix} g_{-1/2} \\ g_{1/2} \\ \vdots \\ \vdots \\ g_{N-1/2} \\ g_{N+1/2} \end{pmatrix}}_{[X]} = \underbrace{\begin{pmatrix} 6(\bar{g}_0) \\ 3(\bar{g}_0 + \bar{g}_1) \\ \vdots \\ \vdots \\ 3(\bar{g}_{N-1} + \bar{g}_N) \\ 6(\bar{g}_N) \end{pmatrix}}_{[B]}$$

3.3 LAG scheme with the Hermite formalism

3.3.1 Hermite formalism using Lagrange interpolation

We outline hereafter the Hermite formulation for the LAG scheme. In order to set the distribution function values at the faces $g_{k+1/2}^+$ and $g_{k+1/2}^-$, we start from the Lagrange interpolation of the flux and then, we make some calculations to show the equivalence with the Hermite formulation.

We build the primitive or cumulative function G at the characteristic feet:

$$\begin{aligned} g_i^{n+1} &= G(x_{i+1/2}^*) - G(x_{i-1/2}^*) \\ G(x_{i+1/2}^*) &= \int_{x_{-1/2}}^{x_{i+1/2}^*} g^n(x) dx. \end{aligned}$$

The value of $G(x_{i+1/2}^*)$ is obtained by an interpolation with third-order Lagrange polynoms of values at cell faces $G(x_{j+1/2})$:

$$G(x) = \sum_{j=i-1}^{i+2} G_{j-1/2} L_j(x), x \in [x_{i-1/2}, x_{i+1/2}]$$

where L_j are the Lagrange polynoms defined as:

$$L_j(x) = \prod_{k=i-1, k \neq j}^{i+2} \frac{x - x_{k-1/2}}{x_{j-1/2} - x_{k-1/2}}$$

and $G_{j-1/2}$ is the value of G at the cell face $j-1/2$.

Some calculations (given in annex sec. 7.1) lead to the expressions (23) and (24) for $G(x_{i+1/2}^*)$:

- if $\beta_{i+1/2} > 0$,

$$\begin{aligned}
G(x_{i+1/2}^*) &= \beta^3(1/6G_{i-3/2} - 1/2G_{i-1/2} + 1/2G_{i+1/2} - 1/6G_{i+3/2}) \\
&+ \beta^2(1/2G_{i-1/2} - G_{i+1/2} + 1/2G_{i+3/2}) \\
&+ \beta(-1/6G_{i-3/2} + G_{i-1/2} - 1/2G_{i+1/2} - 1/3G_{i+3/2}) \\
&+ G_{i+1/2}
\end{aligned} \tag{23}$$

- if $\beta_{i+1/2} < 0$,

$$\begin{aligned}
G(x_{i+1/2}^*) &= \beta^3(1/6G_{i-1/2} - 1/2G_{i+1/2} + 1/2G_{i+3/2} - 1/6G_{i+5/2}) \\
&+ \beta^2(1/2G_{i-1/2} - G_{i+1/2} + 1/2G_{i+3/2}) \\
&+ \beta(1/3G_{i-1/2} + 1/2G_{i+1/2} - G_{i+3/2} + 1/6G_{i+5/2})
\end{aligned} \tag{24}$$

with $G_{k+1/2} = G(x_{k+1/2})$ and the function $\beta_{i+1/2}$:

$$\Delta x \beta_{i+1/2} = x_{i+1/2} - x$$

We omit the subscript for β when the context makes the index obvious.

The values of the primitive function at nodes $G_{k+1/2}$ are related to the cell centred values of the distribution function g_k^n by:

$$g_k^n \Delta x = G_{k+1/2} - G_{k-1/2}.$$

Thus, by replacing the values $G_{k+1/2}$ function of g_k^n in such a way only the primitive function value $G_{i+1/2}$ at cell face $i + 1/2$ remains at the right hand side of relations (23) and (24), we obtain:

- if $\beta_{i+1/2} > 0$,

$$\begin{aligned}
G(x_{i+1/2}^*)/\Delta x &= \beta^3(-1/6g_{i-1}^n + 1/3g_i^n - 1/6g_{i+1}^n) \\
&+ \beta^2(-1/2g_i^n + 1/2g_{i+1}^n) \\
&+ \beta(1/6g_{i-1}^n - 5/6g_i^n - 1/3g_{i+1}^n) \\
&+ G_{i+1/2}/\Delta x
\end{aligned}$$

- if $\beta_{i+1/2} < 0$

$$\begin{aligned}
G(x_{i+1/2}^*)/\Delta x &= \beta^3(-1/6g_i^n + 2/6g_{i+1}^n - 1/6g_{i+2}^n) \\
&+ \beta^2(-1/2g_i^n + 1/2g_{i+1}^n) \\
&+ \beta(-1/3g_i^n - 5/6g_{i+1}^n + 1/6g_{i+2}^n) \\
&+ G_{i+1/2}/\Delta x
\end{aligned}$$

Let us recall the relation (16) between the flux at cell face $i + 1/2$ and the primitive function:

$$\phi_{i+1/2} = G(x_{i+1/2}) - G(x_{i+1/2}^*).$$

Therefore, we have an expression of the flux $\phi_{i+1/2}(\beta)$ function of cell centred values $(\bar{g}_k)_k$:

- if $\beta_{i+1/2} > 0$,

$$\begin{aligned} \frac{\phi_{i+1/2}(\beta)}{\Delta x} &= \beta^3(1/6g_{i-1}^n - 1/3g_i^n + 1/6g_{i+1}^n) \\ &+ \beta^2(1/2g_i^n - 1/2g_{i+1}^n) \\ &+ \beta(-1/6g_{i-1}^n + 5/6g_i^n + 1/3g_{i+1}^n) \end{aligned}$$

- if $\beta_{i+1/2} < 0$

$$\begin{aligned} \frac{\phi_{i+1/2}(\beta)}{\Delta x} &= \beta^3(1/6g_i^n - 2/6g_{i+1}^n + 1/6g_{i+2}^n) \\ &+ \beta^2(1/2g_i^n - 1/2g_{i+1}^n) \\ &+ \beta(1/3g_i^n + 5/6g_{i+1}^n - 1/6g_{i+2}^n) \end{aligned}$$

Let us remember the generic Hermite formulation for the flux (21) function of values at cell faces $g_{k+1/2}^\pm$:

- if $\beta_{i+1/2} > 0$ then

$$\begin{aligned} \frac{\phi_{i+1/2}(\beta)}{\Delta x} &= \beta^3(g_{i-1/2}^+ + g_{i+1/2}^- - 2g_i^n) \\ &+ \beta^2(-g_{i-1/2}^+ - 2g_{i+1/2}^- + 3g_i^n) \\ &+ \beta(g_{i+1/2}^-) \end{aligned}$$

- if $\beta_{i+1/2} < 0$ then

$$\begin{aligned} \frac{\phi_{i+1/2}(\beta)}{\Delta x} &= \beta^3(g_{i+3/2}^- + g_{i+1/2}^+ - 2g_{i+1}^n) \\ &+ \beta^2(2g_{i+1/2}^+ + g_{i+3/2}^- - 3g_{i+1}^n) \\ &+ \beta(g_{i+1/2}^+) \end{aligned}$$

By identifying in the two last expressions of the flux $\phi_{i+1/2}(\beta)$ the coefficients of each degree of the polynomial in variable β , we obtain a necessary expression for values $g_{k-1/2}^\pm$ at cell faces:

$$\begin{cases} g_{k-1/2}^+ = 1/3g_{k-1}^n + 5/6g_k^n - 1/6g_{k+1}^n \\ g_{k+1/2}^- = -1/6g_{k-1}^n + 5/6g_k^n + 1/3g_{k+1}^n. \end{cases} \quad (25)$$

As a conclusion, we have explicit values of $g_{k-1/2}^+$ and $g_{k+1/2}^-$ function of cell centred values of the distribution function at time t^n with (25), so we can directly compute the generic Hermite formulation for the flux (21) for the LAG scheme.

3.3.2 Boundary conditions

The boundary conditions for the LAG scheme are more simple than the PSM ones. Indeed, the coefficient are explicitly depending on the value of the distribution function g_k^n . So we can explicitly set values at boundaries.

Non-periodic boundary conditions: natural conditions

$$\begin{cases} g_{N+3/2}^+ = 1/3g_{N+1}^n + 5/6g_{N+1}^n - 1/6g_{N+1}^n \\ g_{N+3/2}^- = -1/6g_N^n + 5/6g_{N+1}^n + 1/3g_{N+1}^n \\ g_{-1/2}^+ = 1/3g_0^n + 5/6g_0^n - 1/6g_1^n \\ g_{-1/2}^- = -1/6g_0^n + 5/6g_0^n + 1/3g_0^n \end{cases}$$

Periodic boundary conditions

$$\begin{cases} g_{-1/2}^+ = 1/3g_{N-1}^n + 5/6g_0^n - 1/6g_1^n \\ g_{-1/2}^- = -1/6g_{N-2}^n + 5/6g_{N-1}^n + 1/3g_0^n \\ g_{N-1/2}^+ = 1/3g_{N-1}^n + 5/6g_0^n - 1/6g_1^n \\ g_{N-1/2}^- = -1/6g_{N-2}^n + 5/6g_{N-1}^n + 1/3g_0^n \end{cases}$$

4 Limiters

High-order numerical schemes may create spurious oscillations when stiff profiles occur in the distribution function. As usual in the numerical framework of finite volume schemes, we employ a flux limiter to lower these oscillations. The principle of the limiter is to introduce numerical diffusion when stiff profiles are detected, by modifying the flux at the cell faces. We will test limiters found in the literature, the so called Oscillation Limiter (OSL) described in [4], and the one proposed in this report that we call Slope Limited Splines (SLS). These limiters are mainly provided with the PSM scheme, but some might be used for any finite volume scheme.

4.1 ENTropic flux limiter (ENT)

The principle of this limiter proposed in [1] is to make degenerate the fourth order PSM scheme to a second order centred flux to reduce the anti-diffusive behaviour of the scheme when it occurs. The position where anti-diffusion occurs is detected looking at the second order equivalent equation solved by the scheme, obtained by a Taylor expansion. This equivalent equation shows a diffusion term at second order, which the sign should be positive, leading to numerical diffusion and thus stability. When this sign is negative, the scheme is anti-diffusive and it is replaced by a centred scheme. The corresponding algorithm is the following:

Algorithm

- Computation of the PSM flux $\phi_{i+1/2}^{PSM}$ and the centred flux $\phi_{i+1/2}^{CEN} = \alpha_{i+1/2} \frac{\bar{g}_i^n + \bar{g}_{i+1}^n}{2}$.
- if $(\phi_{i+1/2}^{CEN} - \phi_{i+1/2}^{PSM})(\bar{g}_{i+1}^n - \bar{g}_i^n) > 0$ then the flux $\phi_{i+1/2}^{PSM}$ is supposed to be diffusive.
- if $(\phi_{i+1/2}^{CEN} - \phi_{i+1/2}^{PSM})(\bar{g}_{i+1}^n - \bar{g}_i^n) < 0$ then the flux $\phi_{i+1/2}^{PSM}$ is supposed to be anti-diffusive, thus it is switched with the centred flux:

$$\phi_{i+1/2}^{PSM} = \phi_{i+1/2}^{CEN}$$

4.2 UMEDA's limiter (UMEDA)

The 4D advection equation is split in four 1D advection equations. Although the maximum principle is satisfied by the 4D equation, this principle is not fulfilled for each 1D equation. Therefore the extrema of the distribution function (minimum and maximum) are unknown in 1D. The PFC limiter [7] employ these extrema, which are not known in the context of a directional splitting. Therefore, following *Umeda* [15], we modify the extrema definition to get a limiter working in 4D. The Umeda's limiter have been written with Lagrange polynoms.

Algorithm

- We evaluate $g_{min1}, g_{min2}, g_{max1}, g_{max2}$

$$\begin{cases} g_{min1} = \max [\max (\bar{g}_{i-1}^n, \bar{g}_i^n), \min (2\bar{g}_{i-1}^n - \bar{g}_{i-2}^n, 2\bar{g}_i^n - \bar{g}_{i+1}^n)] \\ g_{min2} = \max [\max (\bar{g}_{i+1}^n, \bar{g}_i^n), \min (2\bar{g}_{i+1}^n - \bar{g}_{i+2}^n, 2\bar{g}_i^n - \bar{g}_{i-1}^n)] \\ g_{max1} = \min [\min (\bar{g}_{i-1}^n, \bar{g}_i^n), \max (2\bar{g}_{i-1}^n - \bar{g}_{i-2}^n, 2\bar{g}_i^n - \bar{g}_{i+1}^n)] \\ g_{max2} = \min [\min (\bar{g}_{i+1}^n, \bar{g}_i^n), \max (2\bar{g}_{i+1}^n - \bar{g}_{i+2}^n, 2\bar{g}_i^n - \bar{g}_{i-1}^n)] \end{cases}$$

- We set g_{min}, g_{max}

$$\begin{cases} \bar{g}_{min} = \max [0, \min (g_{min1}, g_{min2})] \\ \bar{g}_{max} = \max [g_{max1}, g_{max2}] \end{cases}$$

- We define L_i^+ and L_i^-

$$L_i^+ = \begin{cases} \text{if } \bar{g}_{i+1} - \bar{g}_i \geq 0, \min(2(\bar{g}_i - \bar{g}_{min}), \bar{g}_{i+1} - \bar{g}_i) \\ \text{if } \bar{g}_{i+1} - \bar{g}_i < 0, \max(2(\bar{g}_i - \bar{g}_{max}), \bar{g}_{i+1} - \bar{g}_i) \end{cases}$$

$$L_i^- = \begin{cases} \text{if } \bar{g}_i - \bar{g}_{i-1} \geq 0, \min(2(\bar{g}_{max} - \bar{g}_i), \bar{g}_i - \bar{g}_{i-1}) \\ \text{if } \bar{g}_i - \bar{g}_{i-1} < 0, \max(2(\bar{g}_{min} - \bar{g}_i), \bar{g}_i - \bar{g}_{i-1}) \end{cases}$$

- We finally redefine the flux:

$$G_{i+1/2}(x) = \bar{g}_i + x(1-x)(2-x)(L_i^+/6) + x(1-x)(1+x)(L_i^-/6)$$

We obtain the LAG reconstruction without limiter by setting:

$$\begin{cases} L_k^+ = \bar{g}_{k+1} - \bar{g}_k \\ L_k^- = \bar{g}_k - \bar{g}_{k-1} \end{cases}$$

4.3 OScillation Limiter (OSL)

The OScillation Limiter (OSL) proposed in [4] is really using the Hermite formalism. It compares the values at cell faces $g_{k-1/2}^\pm$ obtained with the LAG and the PSM schemes to the value computed with a linear reconstruction. This later consists in computing an average at the face of the left and right cells centred values. If the values computed with PSM and LAG are not simultaneously upper or lower than the average value, then we take the average value. If not, a mixed scheme between PSM, LAG and the average reconstruction is performed. The OSL limiter includes a parameter $C > 1$ determining the proportion in the average of PSM and LAG fluxes in the limited flux.

Algorithm

- Computation of $g_{k-1/2}^\pm$ values for both PSM et LAG schemes.
- Average value at the $k - 1/2$ and $k + 1/2$ nodes: $g_{k-1/2}^{ave} = (\bar{g}_k^n + \bar{g}_{k-1}^n)/2$ and $g_{k+1/2}^{ave} = (\bar{g}_k^n + \bar{g}_{k+1}^n)/2$.

- The following formula perform the choice for the limited flux according to the regularity of the cell faces values:

$$\begin{aligned}
& - \text{ if } (g_{k-1/2,LAG}^+ - g_{k-1/2}^{ave})(g_{k-1/2,PSM} - g_{k-1/2}^{ave}) > 0 \text{ then} \\
& \quad g_{k-1/2}^+ = g_{k-1/2}^{ave} \text{ else} \\
& \quad g_{k-1/2}^+ = g_{k-1/2}^{ave} \\
& \quad + \text{sign}(g_{k-1/2,PSM} - g_{k-1/2}^{ave}) \\
& \quad \min(C|g_{k-1/2,LAG}^+ - g_{k-1/2}^{ave}|, |g_{k-1/2,PSM} - g_{k-1/2}^{ave}|)
\end{aligned} \tag{26}$$

$$\begin{aligned}
& - \text{ if } (g_{k+1/2,LAG}^- - g_{k+1/2}^{ave})(g_{k+1/2,PSM} - g_{k+1/2}^{ave}) > 0 \text{ then} \\
& \quad g_{k+1/2}^- = g_{k+1/2}^{ave} \text{ else} \\
& \quad g_{k+1/2}^- = g_{k+1/2}^{ave} \\
& \quad + \text{sign}(g_{k+1/2,PSM} - g_{k+1/2}^{ave}) \\
& \quad \min(C|g_{k+1/2,LAG}^- - g_{k+1/2}^{ave}|, |g_{k+1/2,PSM} - g_{k+1/2}^{ave}|)
\end{aligned} \tag{27}$$

4.4 Slope Limited Splines (SLS)

The limiting procedure basically aims to cut the oscillations generated by strong gradients in the distribution function profile, where high order schemes will produce overshoots and spurious oscillations. We propose here to measure these gradients and to add diffusion where strong gradients are detected. The diffusion is added by mixing the high order scheme with a first order upwind flux. The more the gradient is steep, the more we raise the proportion of upwind flux in the average with the high order scheme. The evaluation of the gradient is given by a function θ and we estimate the diffusion needed with a function $\gamma(\theta) \in [0, 1]$ based on the minmod like limiter function (fig. 7):

$$\phi_{i+1/2}^{new} = \gamma(\theta_{i+1/2}) \phi_{i+1/2}^{PSM} + (1 - \gamma(\theta_{i+1/2})) \phi_{i+1/2}^{upwind}$$

where

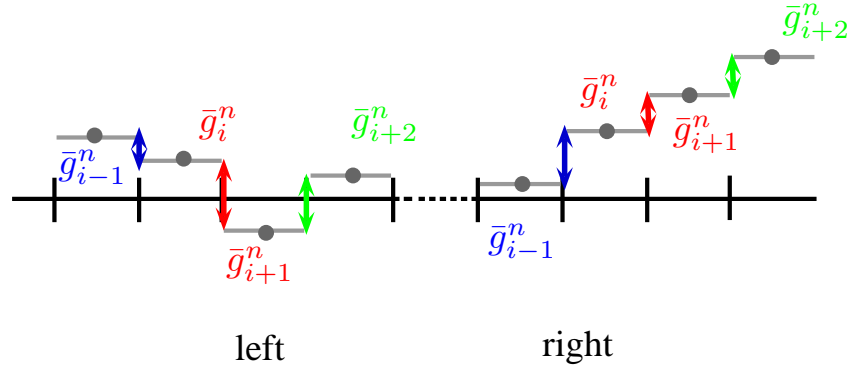
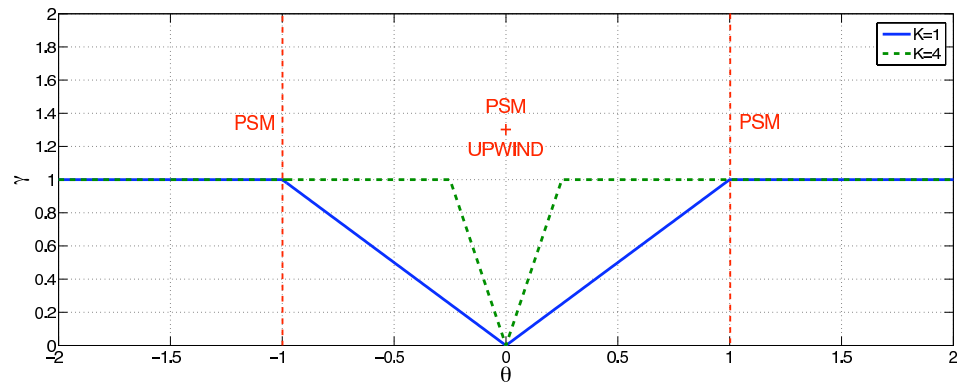
$$\begin{cases} \phi_{i+1/2}^{upwind} = \alpha_{i+1/2} \frac{\bar{g}_i^n + \bar{g}_{i+1}^n}{2} - \text{sign}(\alpha_{i+1/2}) \frac{\bar{g}_{i+1}^n - \bar{g}_i^n}{2} \\ \alpha_{i+1/2} = \Delta t a_{i+1/2} \end{cases}$$

We define $\theta_{i+1/2}$ as the classical slope ratio of the distribution which depends on the direction of the displacement (fig. 6):

$$\theta_{i+1/2} = \begin{cases} \frac{\bar{g}_i^n - \bar{g}_{i-1}^n}{\bar{g}_{i+1}^n - \bar{g}_i^n} & \text{if } \alpha_{i+1/2} > 0 \\ \frac{\bar{g}_{i+2}^n - \bar{g}_{i+1}^n}{\bar{g}_{i+1}^n - \bar{g}_i^n} & \text{if } \alpha_{i+1/2} < 0 \end{cases}$$

However, the classical limiters as minmod $\gamma_{i+1/2} = \max(0, \min(\theta_{i+1/2}, 1))$, set γ to 0 when $\theta < 0$. That means that the scheme turns to order 1 when an extrema exists, i.e. the slope ratio $\theta < 0$. These extrema are thus quickly diffused and that leads to loose the benefits of a high order method. For SLS, the choice is to let the high-order scheme deal with the extrema and only add diffusion when high gradients occurs, i.e. the slope ratio $\theta \approx 0$. We also introduce a constant K in relation to control the maximum slope allowed without adding diffusion. The SLS limiter function is thus to set $\gamma = 1$ for any values of θ , except close to $|\theta| = 0$ where strong gradients occurs, see figure 7:

$$\gamma_{i+1/2} = \max(0, \min(K|\theta_{i+1/2}|, 1)) \tag{28}$$

Figure 6: Computation of $\theta_{i+1/2}$ Figure 7: γ function for the SLS limiter

5 Numerical results

We consider both 1D and 4D test cases. The 1D advection test case is performed with a constant velocity to give a qualitative view on the effects of each limiter. We are particularly interested in advection of profiles with strong gradients, as discontinuous initial distribution functions as a step. However, we have to keep in mind that this situation of a discontinuous shape advected with a constant velocity field does not occur in Vlasov 4D drift-kinetic simulations for many reasons: the advections are not constant, the long time constant advection configuration does not appear, and discontinuous functions (steps) do not exist in the Vlasov model. However, it is relevant to investigate advection of discontinuous functions since Vlasov models leads to stretched structures in the flow that makes appear strong gradients, which could be assimilated to discontinuities.

The 4D drift-kinetic test case is performed with the 5D Gysela code following the benchmark presented in [9]. This benchmark is to simulate instabilities growing in the plasma leading to shapes like filaments and vortex. We compare the results to evaluate the limiters effects on the development of turbulent structures. We will focus on two directions, (r, θ) , among the four $(r, \theta, \phi, v_{\parallel})$ since strong gradients of the distribution function appear mainly in (r, θ) planes (further details in [1]). We give algorithms used in Gysela to solve the advection equation in sec. 7.4.

5.1 Tools

We propose hereafter some quantitative tools to compare the efficiency of each limiter. The L^2 norm and the entropy may be used to investigate the diffusivity of each limiter, since each should theoretically be kept constant. The total energy conservation is related to the treatment of small structures provided by the instabilities. In addition, we also introduce the total variation (TV) norm to estimate the numerical oscillations created by the scheme.

Total Variation

The total variational norm (TV) is used to estimate the rate of oscillations produced by a scheme, compare to another.

- In **1D**:

$$TV(g(t)) = \int_{x_{min}}^{x_{max}} \left| \frac{dg(t, x)}{dx} \right| dx$$

$$TV(g^n) = \frac{1}{\Delta x} \sum_{i=0}^{N_x} |\bar{g}_{i+1}^n - \bar{g}_i^n|$$

- In **4D**, we restrict the diagnostic to an (r, θ) plane:

$$TV(g(t)) = \int_{r_{min}}^{r_{max}} \int_0^{2\pi} |\nabla_{r, \theta} g(t, r, \theta)|_{L^2} dr d\theta$$

$$TV(g^n) = \sum_{i=0}^{N_r} \sum_{j=0}^{N_\theta} \sqrt{\left(\frac{\bar{g}^n(i+1, j) - \bar{g}^n(i, j)}{\Delta r} \right)^2 + \left(\frac{\bar{g}^n(i, j+1) - \bar{g}^n(i, j)}{\Delta \theta} \right)^2}$$

L^2 norm and entropy

The L^2 norm is defined as:

$$L^2(f(t)) = \int (f(t, X, v_{\parallel}))^2 dX dv_{\parallel}, \quad dX = r dr d\theta d\phi.$$

The entropy measures the information created or destroyed by a phenomena. We can thus expect that a raising entropy estimates the information lost by numerical diffusion.

$$S(f(t)) = - \int f(t, X, v_{\parallel}) \log(f(t, X, v_{\parallel})) dX dv_{\parallel}, \quad dX = r dr d\theta d\phi$$

Total Energy

The conservation of the total energy ϵ_{tot} writes as the sum of the kinetic energy ϵ_{kin} and the potential energy ϵ_{pot} defined as follow:

$$\epsilon_{tot} = \epsilon_{kin} + \epsilon_{pot} = \int \frac{1}{2} m_i v_{\parallel}^2 (f - f_{eq}) dV dv_{\parallel} + \frac{1}{2} \int e \phi (n_i - n_0) dX.$$

For further details, see [9].

Quality factor

We propose here to define a quality factor to establish a quantitative evaluation of the limiters efficiency. Since the L^2 norm should be conserved by the Vlasov equation, it furnishes a good estimation of the diffusive behaviour of a scheme, because it decreases this norm. On the other hand, the TV norm estimates the numerical oscillations, or also the diffusion in a way, but more 'locally'. The quality of a scheme may be defined as its ability to limit the oscillations (small TV) with the less numerical diffusion (high L^2 norm). Therefore we introduce the quality criterion Q defined as:

$$Q(f(t)) = \frac{L^2(f(t))}{TV(f(t))} \quad (29)$$

5.2 Test of the limiters on constant 1D advections

We test the limiters in 1D on a step function ¹. The step exhibits the problems for which the limiters are required. This 1D benchmark gives a first overview on the limiters capabilities.

The benchmark consists in solving a constant advection on a periodic domain divided in 80 cells. The displacement is set to 0.2 cell per iteration. The indicated times are the number of iterations.

¹the distribution function f is set to 1 on a part of the domain else to 0

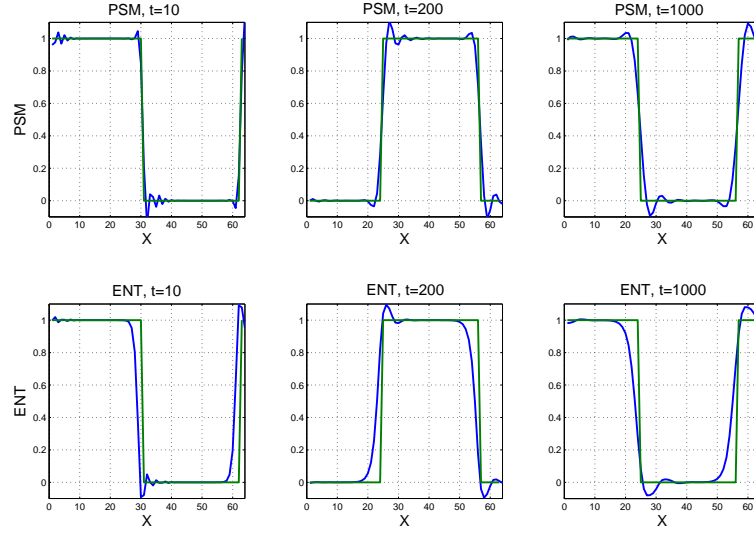


Figure 8: Constant advection on a step with the PSM scheme and the PSM scheme with entropic limiter ENT. The domain is meshed on 80 cells with periodic boundary conditions and the displacement is set to 0.2 cell per iteration.

PSM-ENT We observe on figure 8 that the ENT limiter cuts off the spurious oscillations of the PSM scheme at the left side of the discontinuity. However, at the right side, oscillations are weakly damped.

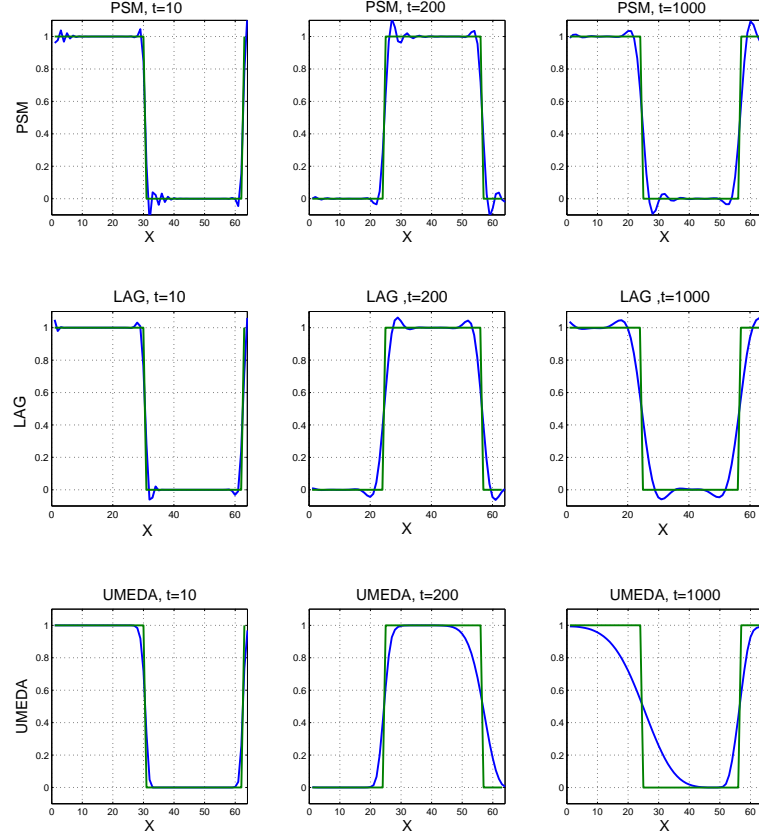


Figure 9: Constant advection on a step with the LAG scheme and the LAG scheme with the Umeda limiter. The domain is meshed on 80 cells with periodic boundary conditions and the displacement is set to 0.2 cell per iteration.

LAG-UMEDA We observe on figure 9 that the LAG scheme is much more diffusive than the PSM scheme without really better satisfying a maximum principle. The under/overshoots of the LAG scheme are cut off by the Umeda limiter. Although the LAG scheme with Umeda limiter respects a maximum principle, the step signal is quickly diffused and it seems not appropriate for long time simulations.

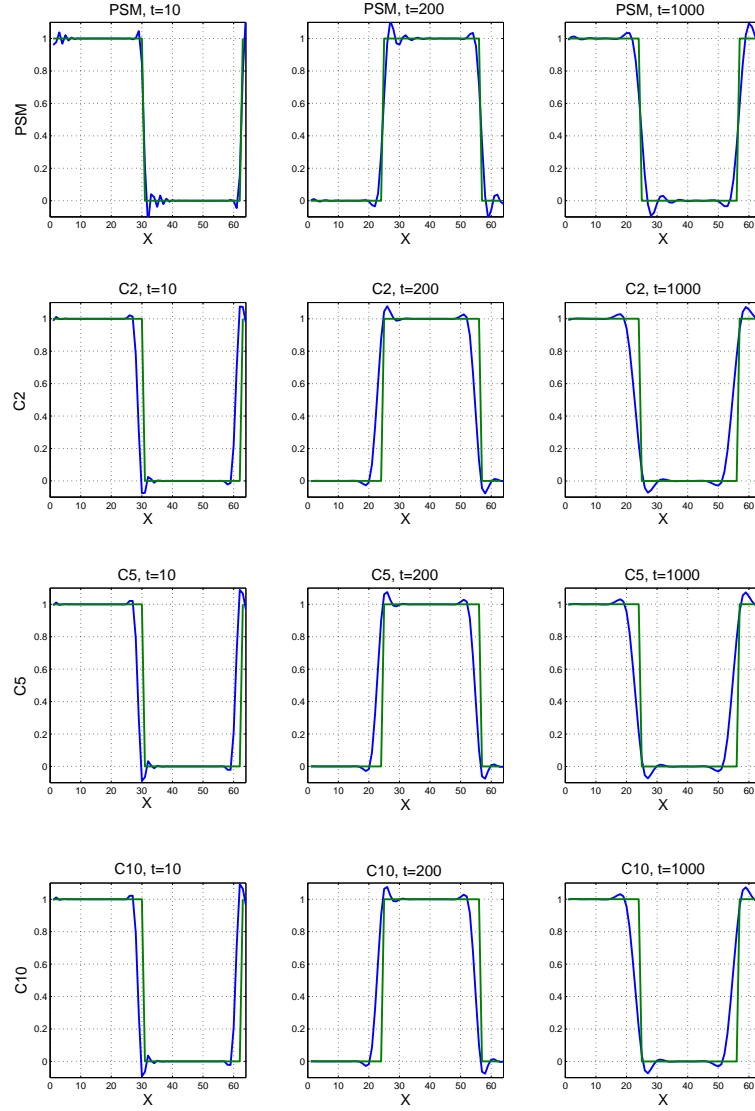


Figure 10: Constant advection on a step with the PSM scheme and the PSM scheme with the OSL limiter with different values of parameter C . The domain is meshed on 80 cells with periodic boundary conditions and the displacement is set to 0.2 cell per iteration.

PSM-OSL We observe on figure 10 that the OSL limiter reduces the oscillations without introducing much diffusion. However, there is still under/overshoots at both sides of the discontinuity and a slight offset of the solution occurs compare to the exact solution, but it seems not increase with time. The constant C of the OSL limiter has almost no influence on the results.

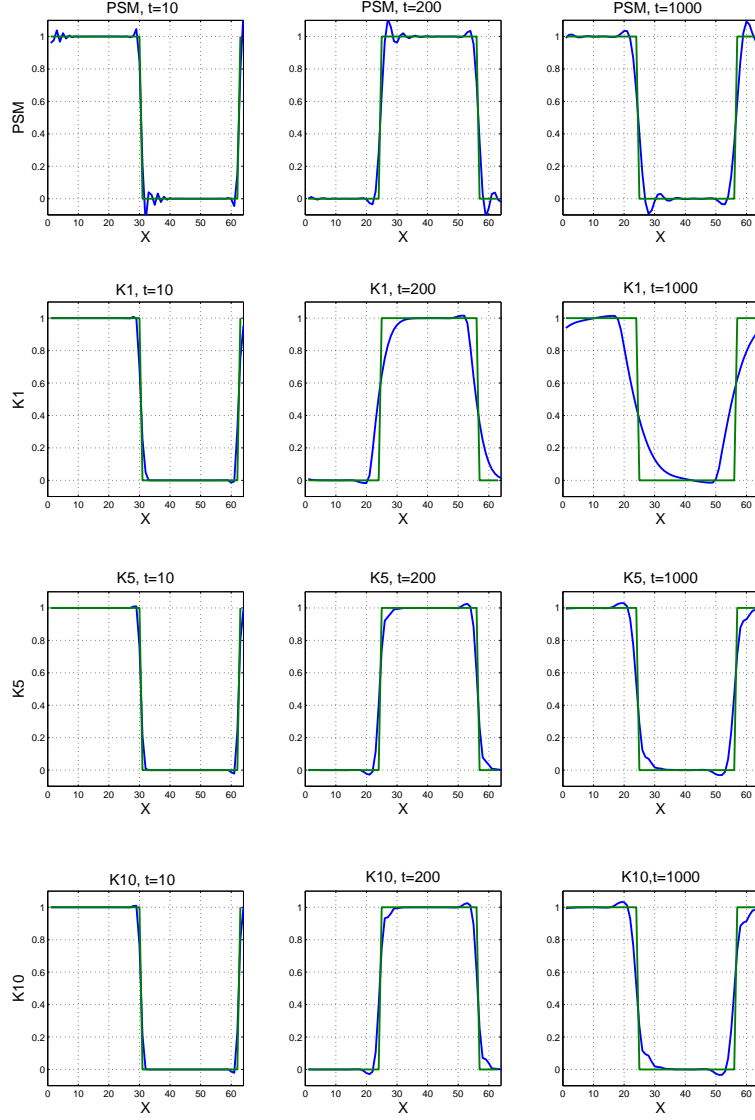


Figure 11: Constant advection on a step with the PSM scheme and the PSM scheme with the SLS limiter with different values of parameter K . The domain is meshed on 80 cells with periodic boundary conditions and the displacement is set to 0.2 cell per iteration.

PSM-SLS We observe on figure 11 that the SLS limiter cuts off oscillations at the right side of the discontinuity and keeps the overshoot at left side. The results of SLS limiter with $K = 1$ is too much diffusive (equivalent to a minmod limiter). With $K = 5$, the result is almost the same as with $K = 10$ and the accuracy at discontinuities (slope of the solution at discontinuities) is almost the same as with the PSM scheme.

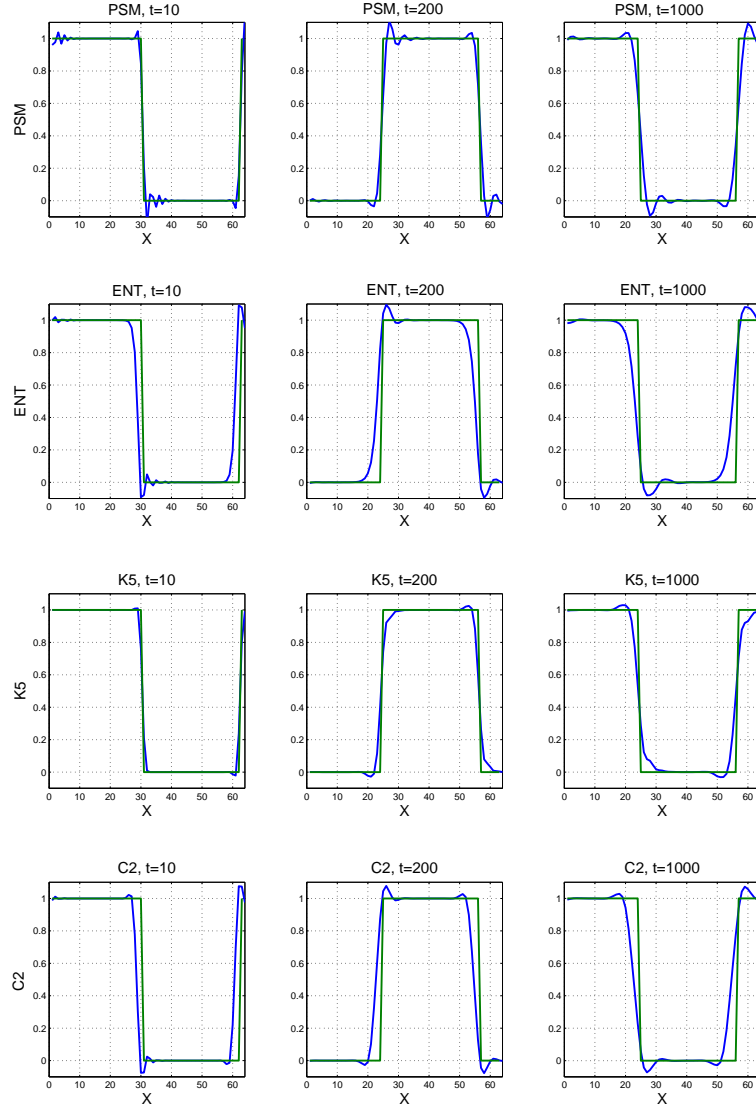


Figure 12: Constant advection on a step with the PSM scheme, the PSM scheme with the ENT limiter, the SLS limiter with parameter $K = 5$ and the OSL limiter with parameter $C = 2$. The domain is meshed on 80 cells with periodic boundary conditions and the displacement is set to 0.2 cell per iteration.

Comparison of all limiters for PSM It is interesting to observe the action of each different limiter for PSM on the oscillations. The ENT limiter cuts off the oscillations at left side of the discontinuities, the SLS $K = 5$ limiter cuts off the oscillations at right side of the discontinuities and the OSL limiter reduces the oscillations at both sides.

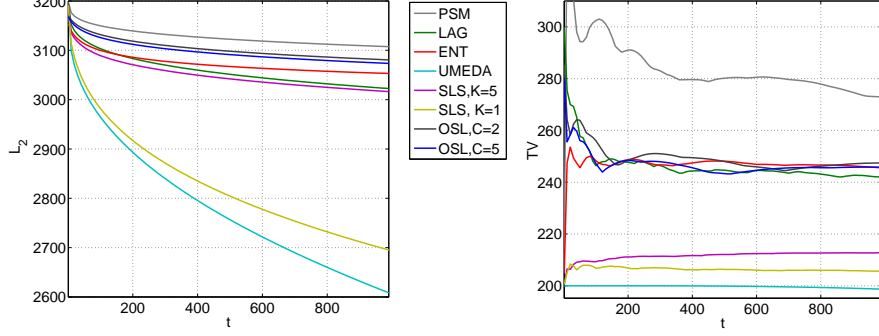


Figure 13: L^2 norm (left) and total variation (right) as a function of the time for the different methods applied to a step with 1D constant advection on a periodic domain.

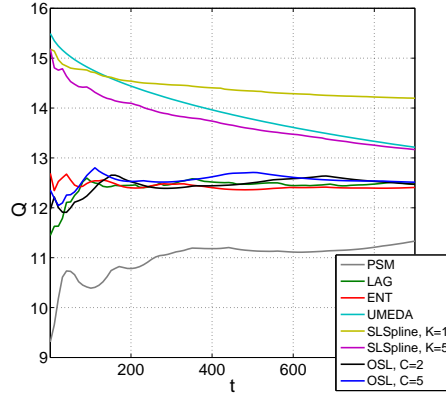


Figure 14: Quality factor for all the methods as a function of the time for the different methods applied to a step with 1D constant advection on a periodic domain.

Quality factor The quality factor (fig. 14), as defined previously in (29) as the ratio of L^2 norm over TV norm, shows three groups of curves:

- On the lower part, the PSM scheme curve shows that even conserving well the L^2 norm, the TV norm is much higher than the other schemes because of spurious oscillations. The quality factor is then poor for PSM.
- The group of four curves at the middle (LAG, PSM-ENT, PSM OSM with $C=2$ and $C=5$) are conserving the L^2 norm with the same order (best PSM-OSL $C=2$, worst LAG) and with almost the same level of oscillations according to the TV norm. The quality factor is thus equivalent for this group of schemes.
- On the upper part, LAG-UMEDA and PSM-SLS $K=1$ (minmod limiter) are very diffusive according to the L^2 norm, so they kill all the oscillations

and the TV norm is very weak. The quality factor is high because these diffusive schemes have too low TV norms and cannot properly be compared with the others. However, the PSM-SLS $K=5$ scheme conserves the L^2 norm in a comparable way with the other schemes, but with a very low TV norm. The quality factor is thus good for this scheme and can be compared with to the other schemes.

As a conclusion for this comparison of all schemes on a linear advection of a step function test case, we could say that the LAG-UMEDA and PSM-SLS $K=1$ (minmod limiter) schemes are too diffusive to be compared with the other schemes considering the quantitative quality factor (29). However, this way of quantitative comparison seems to fit well for this test case with the qualitative comparison or the "feeling" we might have by observing the plotted results in Figure 12.

5.3 4D Drift-kinetic model

In this section, we evaluate the limiter capacities on a 4D drift-kinetic model simulation of instabilities with the Gysela code [9]. The benchmark providing instabilities is described in [1]. We simulate this test case with two mesh resolutions: a low resolution to provide results about all the limiters and a high resolution to get refined results with the best schemes to test also robustness. However, let us first do a quick review of the 4D drift-kinetic model and the time scheme used for high resolution simulations.

We recall hereafter the model described in [8]. The geometrical assumptions of this model for ion plasma turbulence are a cylindrical geometry with coordinates $(r, \theta, z, v_{\parallel})$ and a constant magnetic field $B = B_z e_z$, where e_z is the unit vector in z direction. In this collisionless plasma, the trajectories are governed by the guiding center (GC) trajectories:

$$\frac{dr}{dt} = v_{GC_r}; \quad r \frac{d\theta}{dt} = v_{GC_\theta}; \quad \frac{dz}{dt} = v_{\parallel}; \quad \frac{dv_{\parallel}}{dt} = \frac{q_i}{m_i} E_z \quad (30)$$

with $v_{GC} = (E \times B)/B^2$ and $E = -\nabla\Phi$ with Φ the electric potential. The Vlasov equation governing this system, where the ion distribution function is $f(r, \theta, z, v_{\parallel}, t)$, is the following:

$$\partial_t f + v_{GC_r} \partial_r f + v_{GC_\theta} \partial_\theta f + v_{\parallel} \partial_z f + \frac{q_i}{m_i} E_z \partial_{v_{\parallel}} f = 0. \quad (31)$$

This equation is coupled with a quasi-neutrality equation for the electric potential $\Phi(r, \theta, z)$ that reads:

$$-\nabla_{\perp} \Phi \cdot \left(\frac{n_0(r)}{B \Omega_0} \nabla \Phi \right) + \frac{e n_0(r)}{T_e(r)} (\Phi - \langle \Phi \rangle_z) = n_i - n_0 \quad (32)$$

with $n_i = \int_{v_{\parallel}} f(r, \theta, z, v_{\parallel}) dv_{\parallel}$ and constant in time physical parameters n_0 , Ω_0 , T_e and e .

Let us notice that the 4D velocity field $a = (v_{GC_r}, v_{GC_\theta}, v_{\parallel}, q/m_i E_z)^t$ is divergence free:

$$\nabla \cdot a = \frac{1}{r} \partial_r (r v_{GC_r}) + \frac{1}{r} \partial_\theta (v_{GC_\theta}) + \partial_z v_{\parallel} + \partial_{v_{\parallel}} (q/m_i E_z) = 0$$

because of variable independence $\partial_{v_{\parallel}} E_z = \partial_{v_{\parallel}} (\partial_z \Phi(r, \theta, z)) = 0$, $\partial_z v_{\parallel} = 0$ and

$$v_{GC_r} = \frac{-1}{r B_z} \partial_\theta \Phi \quad \text{and} \quad v_{GC_\theta}/r = \frac{1}{r B_z} \partial_r \Phi,$$

such that

$$\frac{1}{r} \partial_r (r v_{GC_r}) + \frac{1}{r} \partial_\theta (v_{GC_\theta}) = 0.$$

Therefore, one can write an equivalent conservative equation to the preceding Vlasov equation (31):

$$\partial_t f + \partial_r (v_{GC_r} f) + \partial_\theta (v_{GC_\theta} f) + \partial_z (v_{\parallel} f) + \partial_{v_{\parallel}} \left(\frac{q_i}{m_i} E_z f \right) = 0.$$

5.4 Low resolution simulations

The low-resolution simulations run on a $64 \times 128 \times 16 \times 16$ grid ($N_r \times N_\theta \times N_\phi \times N_{v_\parallel}$). The resolution is too low to consider the total energy as a relevant measure. The entropy, L^2 norm and TV norm are used to gauge the limiter's effects. We present in Figures 15 16, 17 three states of the instabilities: the linear phase (left), the beginning of the non-linear phase (center) and the strong non-linear phase (right).

In figure 15, we see the difference of behaviour of the PSM scheme based on a spline reconstruction method and the LAG scheme based on Lagrangian polynoms. The LAG scheme diffuses much more the solution even at early times, and this is accentuated using the LAG-UMEDA scheme which makes disappear even the big structures. However, the PSM scheme solution shows a lot of small structures, actually at the scale of the mesh, which are suspected of coming from spurious oscillations already observed on the 1D step test case. Moreover, these oscillations often violate the theoretical maximum principle of the solution and may lead to crash the simulation.

In figure 16, we see the effect of the ENT limiter on the PSM scheme. It reduces a lot the oscillations with a figure showing much less small structures, but showing a more diffusive behaviour. The PSM-SLS $K=1$ scheme is extremely diffusive and makes disappear even the big structures.

In figure 17, we compare the PSM scheme result with PSM-OSL and PSM-SLS limiters results. Both limiters reduces efficiently the oscillations. The limiter PSM-SLS seems a little more diffusive than PSM-OSL.

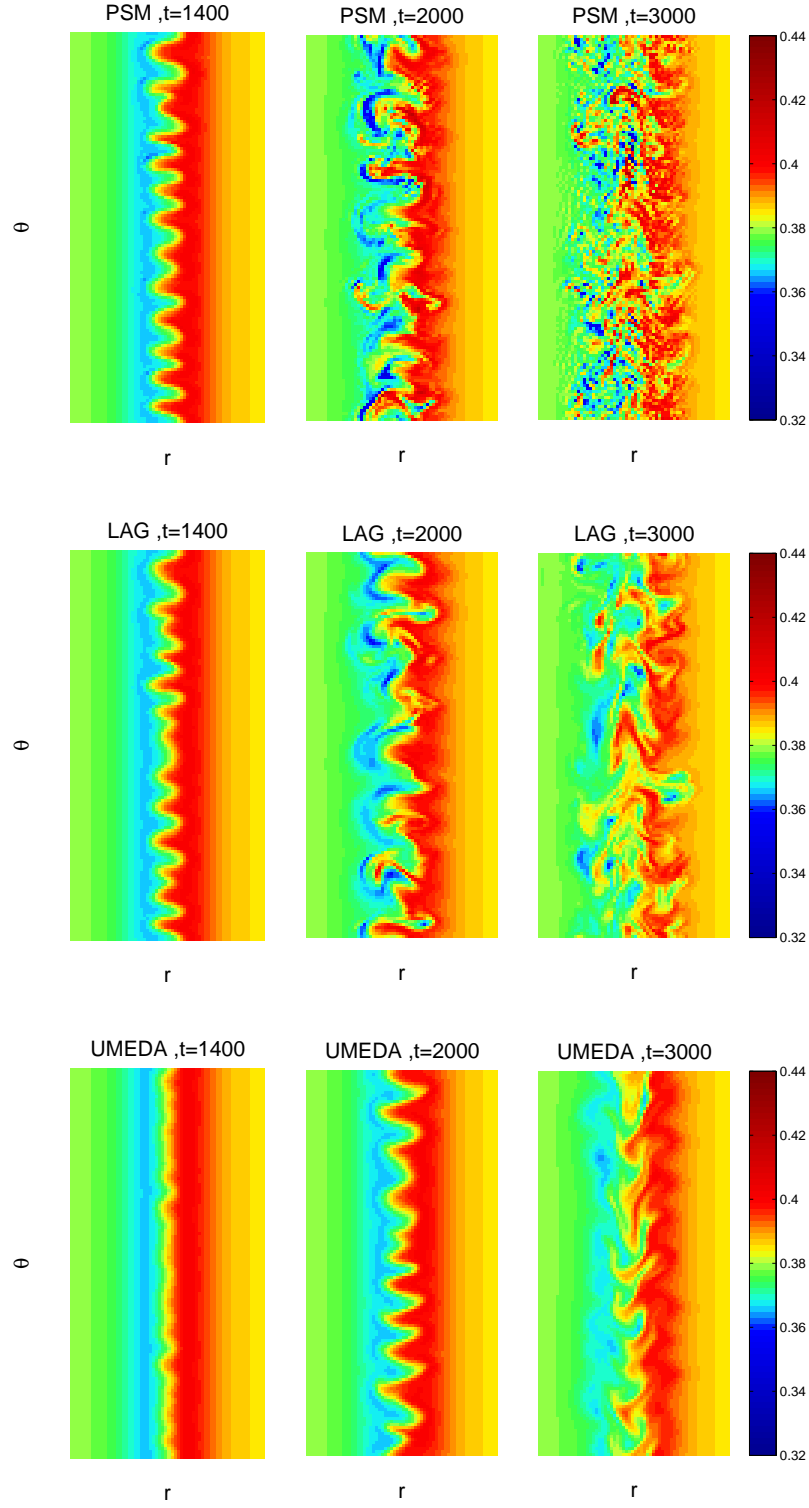


Figure 15: Solution in a (r, θ) plane with $N_r \times N_\theta = 64 \times 128$ cells (low resolution) of a 4D test case with PSM, LAG, LAG-UMEDA methods.

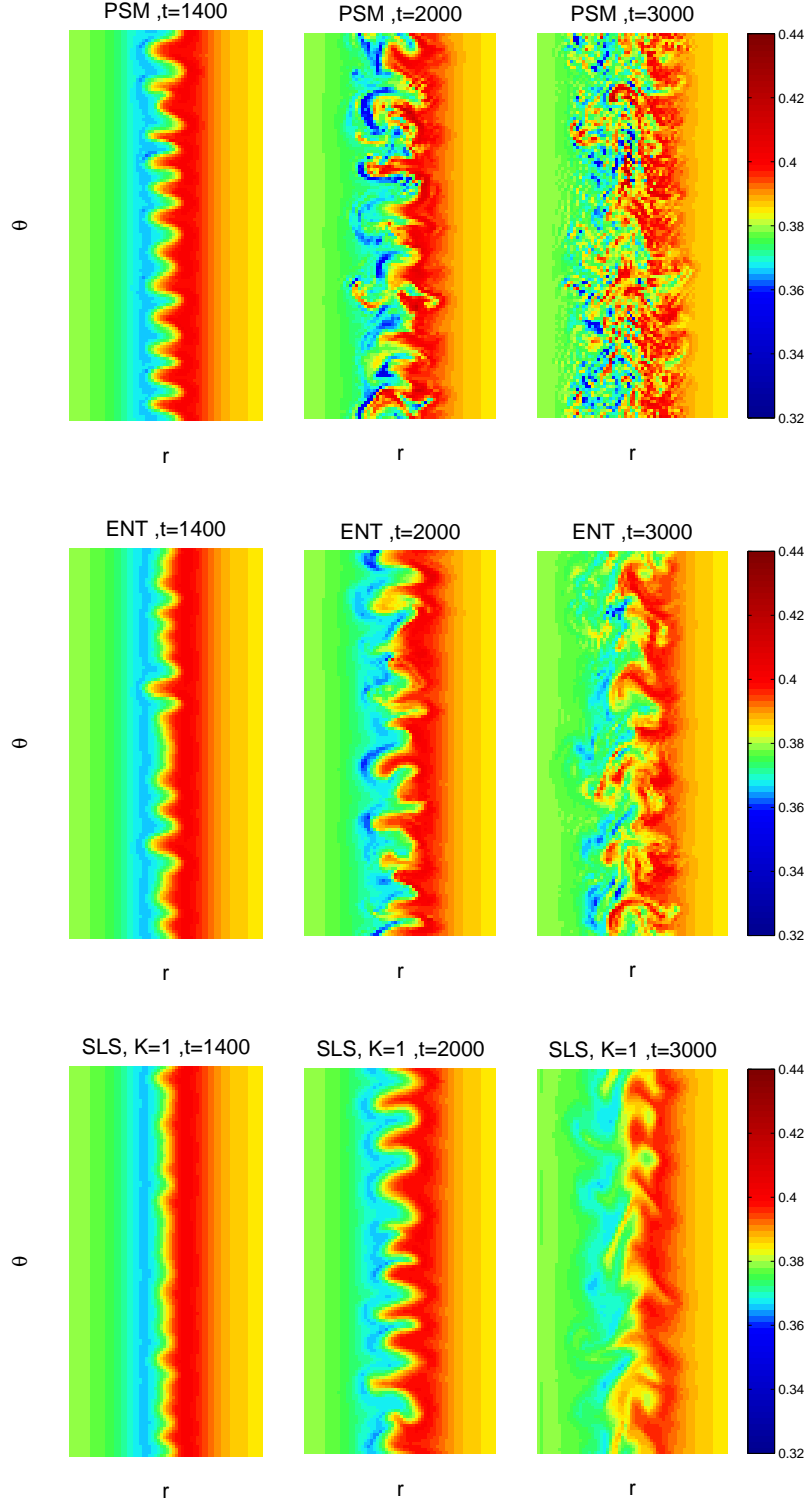


Figure 16: Solution in a (r, θ) plane with $N_r \times N_\theta = 64 \times 128$ cells (low resolution) of a 4D test case with PSM, PSM-ENT and PSM $K=1$ methods.
RR n° 7467

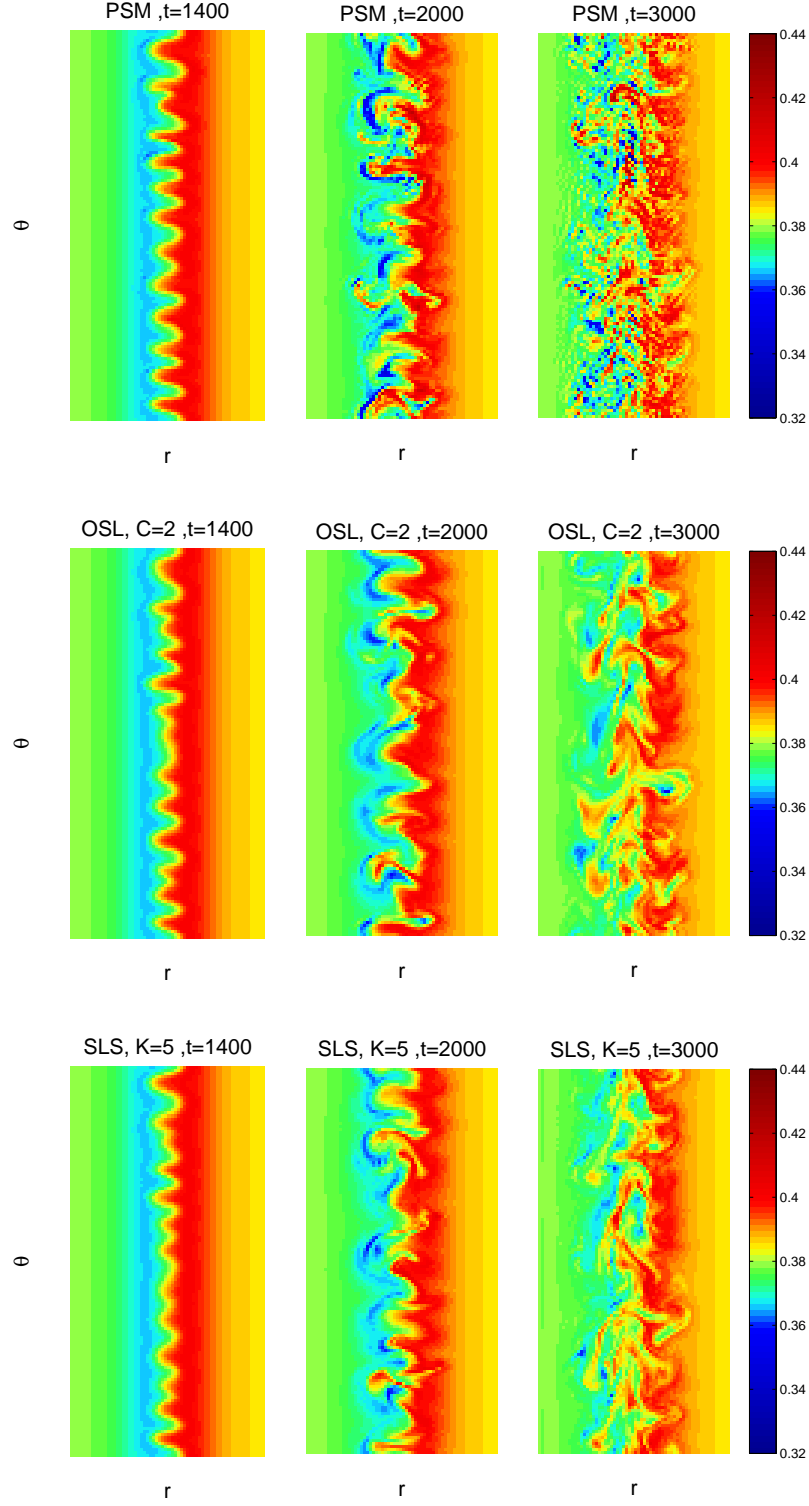


Figure 17: Solution in a (r, θ) plane with $N_r \times N_\theta = 64 \times 128$ cells (low resolution) of a 4D test case with PSM, PSM-OSL C=2 and PSM-SLS K=5 methods.

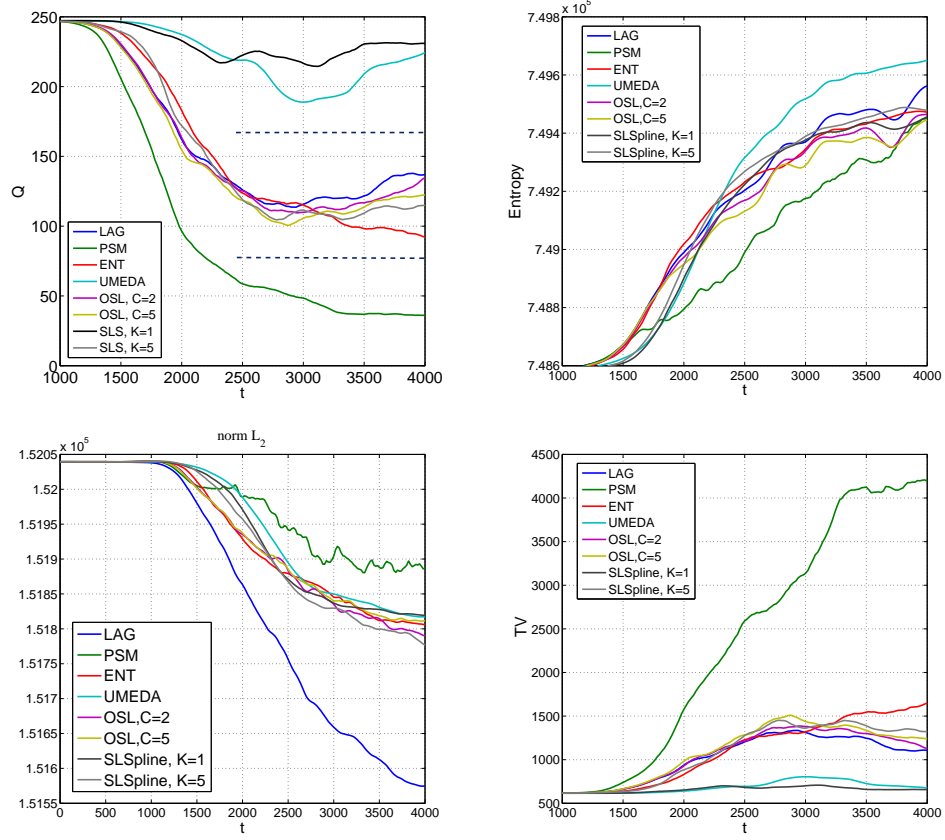


Figure 18: Quality factor (left top), entropy (right top), L^2 norm (left bottom), TV norm (right bottom) for the 4D simulation low resolution $64 \times 128 \times 16 \times 16$ with all the methods presented.

In figure 18 we see the effect of each limiter on the quality factor, entropy, L^2 norm and TV norm.

- The LAG scheme gives very poor results in the conservation of the L^2 norm, but shows a medium quality factor.
- The LAG-UMEDA and PSM-SLS K=1 schemes are still diminishing the TV norm at a very low rate as for the 1D step test case, because they are very diffusive. Even if the quality factor is good and the L^2 norm close to the other schemes, the qualitative results of figure 15 and 16 show that even big structures are not well captured with these schemes.
- The PSM scheme without limiter has still a very different behaviour compare with the others : the L^2 norm and entropy plots show that this scheme is less diffusive and the TV norm plots confirms the qualitative 2D profiles 17 that it is much more oscillating than the other schemes. Thus the quality factor is lower than the other schemes.

- The PSM scheme with limiters ENT, OSL, and SLS K=5 show equivalent medium quality factors and TV norms and conservation of entropy and L^2 norm for this coarse benchmark.

Conclusion of the 4D results with low resolution section

The results obtained for this 4D test case at low resolution confirm those obtained for the 1D linear step test case. The PSM scheme shows oscillations as the TV norm is high compared to the other schemes, but preserves the better L^2 norms and the entropy. The PSM-SLS K=1 and LAG-UMEDA schemes are diffusive and seem experimentally of lower order of accuracy than the other schemes. The PSM-SLS K=1 limiter, PSM-OSL limiter and ENT limiter are showing a similar level of accuracy. The high resolution simulation results following will permit to go further in these investigations.

5.5 High resolution simulation

We propose in this section to compare PSM, OSL and SLS with an high resolution mesh relatively to the previous resolution. The mesh is constituted with $N_r \times N_\theta \times N_\phi \times N_{v_\parallel} = 256 \times 512 \times 32 \times 16$. We consider that this mesh is refined enough such that conservation of the total energy becomes relevant to estimate the limiter quality, what was not the case for the low resolution mesh used in the preceding section 5.4.

We have chosen standard values for PSM limiters according to the preceding results : C=2 for the OSL limiter which depends weakly on this value and K=5 for the SLS limiter which seems to be the minimum value to limit diffusion (see results with K=1 in section 5.4).

The results are presented in three ways : full (r, θ) cut planes, zoom in smaller boxes in these cut planes to better see the influence of the limiters on small structures and 1D diagnostics with the total energy, the quality factor (29), the entropy, the L^2 and TV norms.

Comparison of results in (r, θ) cut planes

In figure 19 and 20, we present at different times (r, θ) profiles of the distribution function in the non-linear phase. The influence of the SLS or OSL limiters is weak at the beginning of the linear phase, i.e. time $t = 2290$. Afterward, a lot of small structures as filaments develop, where steep gradients exist. Therefore for later times, the differences between schemes results increase because of the different actions of SLS and OSL limiters.

- The PSM scheme shows spurious oscillations very quickly in the non-linear phase. As seen in the 1D step test-case and in the 4D test-case with low resolution, the PSM scheme develops oscillations when transporting discontinuities or steep gradients which occurs in this flow where filaments and vortex develop. Unfortunately, these oscillations lead to the crash of the PSM simulation at final time presented $t = 3637$.
- The PSM-SLS scheme does almost develop no oscillations even in the latest time. The price to pay is a qualitatively more diffused profile of distribution function.
- The PSM-OSL scheme does neither develop oscillations at early times of the non-linear phase. The OSL limiter produces less diffusion than the SLS limiter. However, at late times the same kind of oscillations than the PSM scheme develops.

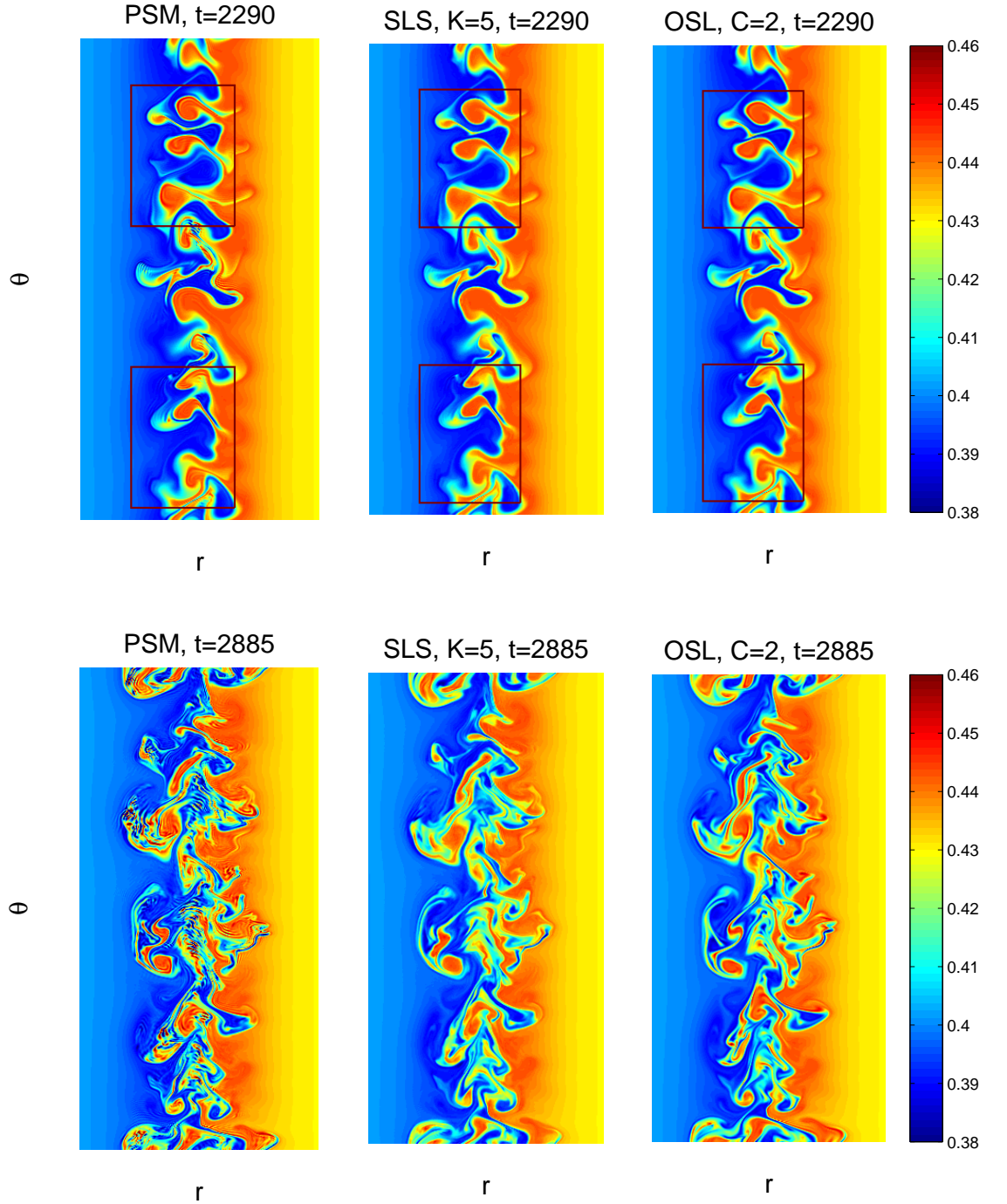


Figure 19: 4D simulations on $N_r \times N_\theta = 256 \times 512$ cells (high resolution) along (r, θ) with PSM (left), (center) SLS, $K=5$ and OSL ($C=2$) (right) at three different times: $t=2290$ (top), $t=2885$ (bottom).

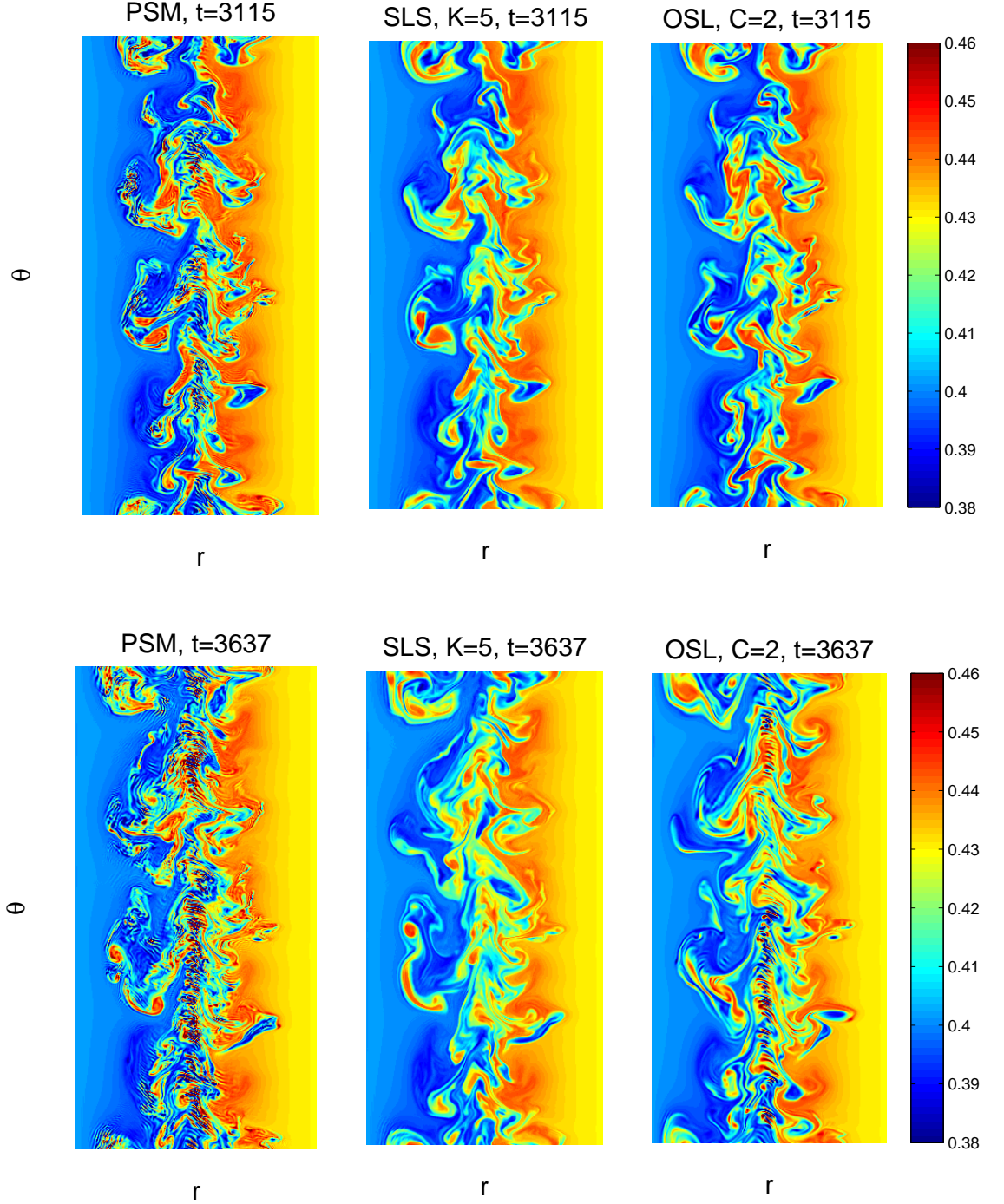
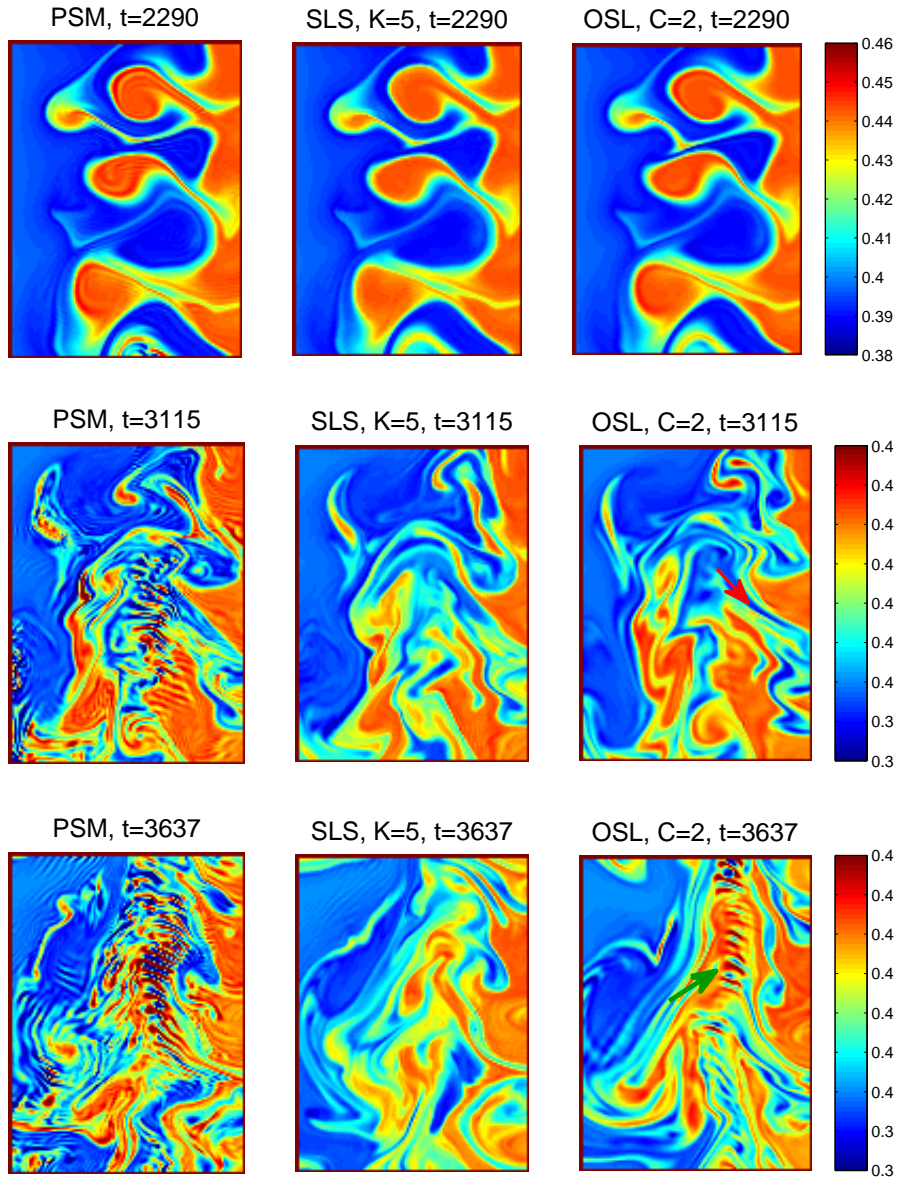


Figure 20: 4D simulations on $N_r \times N_\theta = 256 \times 512$ cells (high resolution) along (r, θ) with PSM (left), SLS, $K=5$ (center) and OSL, $C=2$ (right) at three different times: $t = 3115$ (top), $t = 3637$ (bottom).

Figure 21: Zoom on the zone A of the high-resolution (r, θ) profiles (fig. 19)

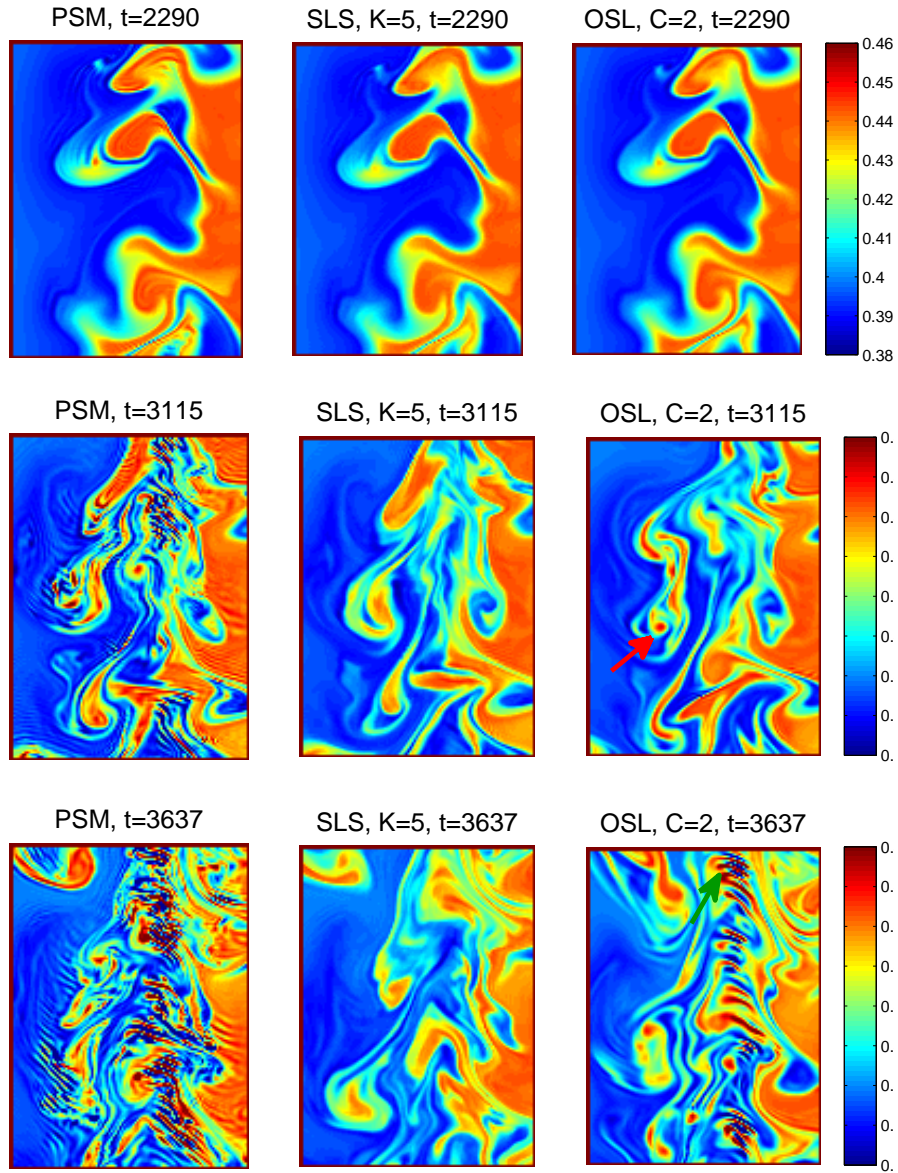


Figure 22: Zoom on a the zone B of the high-resolution (r, θ) profiles (fig. 19)

Comparison of 1D integrated quantities We present in figure 23 the time evolution of the total energy, the entropy, the L^2 and TV norms and the quality factor (29). The results are for the PSM scheme with no limiter, the PSM-SLS scheme with $K=5$ and $K=10$ and the PSM-OSL scheme with $C=1$ and $C=2$.

- The PSM scheme conserves the total energy the better until the simulation crashes down. Entropy and L^2 norm show that this scheme is the less diffusive but with the highest TV norm, which comes from the spurious oscillations seen on (r, θ) profiles and leads to a bad quality factor.
- On the contrary of what expected looking at (r, θ) profiles, the entropy and L^2 norm show that OSL limiter leads to a little more dissipation than the SLS limiter. The total energy is a little better conserved with the SLS limiter than with the OSL limiter at the beginning of the simulation, but it constantly decreases while the total energy obtained with the OSL limiter stabilise at late times.
- The quality factor is a little better with PSM-SLS than with PSM-OSL. However, both achieve the objective of stabilising the PSM scheme oscillations.

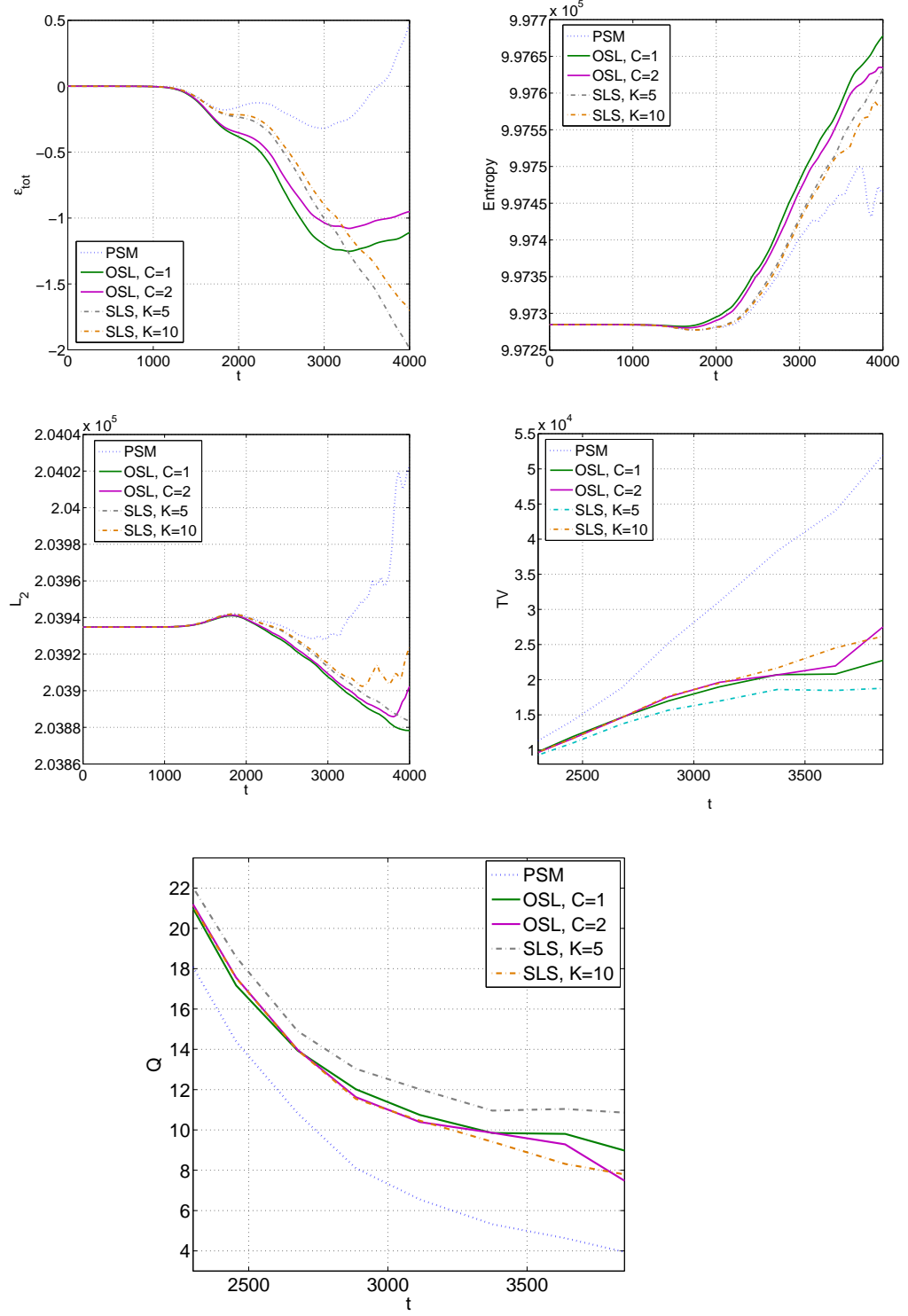


Figure 23: Total energy (left top), entropy (right top), norm L^2 (left bottom) and TV norm (right bottom) for the 4D simulations (high resolution), quality factor Q on the lower part.

6 Conclusion

In this paper, we have investigated conservative schemes for a Vlasov 4D drift-kinetic model for plasma turbulence. These schemes should be accurate to bear long time simulations, so we investigate fourth order schemes. However, high order schemes usually experience difficulties when dealing with steep gradients in the flow, which happens when turbulence develops thin structures or vortices. In particular, spurious oscillations may appear which are not damped by Vlasov models.

We thus have tested limiters to cut off these oscillations. It consists in adding diffusion to the high order scheme at locations where oscillations may appear, i.e. at steep gradients. Then two questions show up: how to introduce diffusion in the scheme and how to detect the location where diffusion is needed. Of course, it should be done efficiently without loosing the accuracy of the scheme by introducing too much diffusion. We have tested the LAG and PSM schemes and the Umeda, ENT, OSL, SLS limiters, which use quite different strategies. We have tested all the schemes on a 1D step linear advection test-case and on a 4D drift-kinetic model test-case.

Another question is how to quantify the action of limiters and to propose a quantitative way of comparison among them. In a classical way, we have checked the conservation of the Vlasov equation invariants as the L^2 norm, the entropy and the total energy for the drift-kinetic model. Even if these quantities permit to compare the behaviour of the schemes, they do not provide a quantitative way to tell one scheme is better than another. We propose an attempt for a quality factor Q (29), based on the dissipation measured with the L^2 norm and spurious oscillations measured with the total variation TV norm. It gives a clear answer the question which is the best of two schemes, but obviously it is still subjective because of the choice of quality factor itself. However, comparison of the schemes results with this quality factor is in good agreement with intuition when looking at results graphs in 1D, but in 4D it helps to evaluate the balance between diffusion and spurious oscillations. At the end of this work, we will not conclude that one scheme is better than another, we just have given these results and tools to help the reader to decide by himself.

7 Annexes

7.1 Annexe A: Detailed LAG reconstruction

We give hereafter detailed calculations for the LAG reconstruction.

First, we build the primitive function G at the characteristic feet:

$$g_i^{n+1} = G(x_{i+1/2}^*) - G(x_{i-1/2}^*)$$

$$G(x_{i+1/2}^*) = \int_{x_{-1/2}}^{x_{i+1/2}^*} g_{j(i)}^n(x) dx$$

$G(x_{i+1/2}^*)$ is given by the interpolation with Lagrange polynoms:

$$G(x) = \sum_{j=i-1}^{i+2} G_{j-1/2} L_j(x), x \in [x_{i-1/2}, x_{i+1/2}]$$

where L_j are the Lagrange polynoms defines as:

$$L_j(x) = \prod_{k=i-1, k \neq j}^{i+2} \frac{x - x_{k-1/2}}{x_{j-1/2} - x_{k-1/2}}$$

Hence,

$$L_{i-1}(x) = \prod_{k=i-1, k \neq i-1}^{i+2} \frac{x - x_{k-1/2}}{x_{i-3/2} - x_{k-1/2}}$$

$$L_i(x) = \prod_{k=i-1, k \neq i}^{i+2} \frac{x - x_{k-1/2}}{x_{i-1/2} - x_{k-1/2}}$$

$$L_{i+1}(x) = \prod_{k=i-1, k \neq i+1}^{i+2} \frac{x - x_{k-1/2}}{x_{i+1/2} - x_{k-1/2}}$$

$$L_{i+2}(x) = \prod_{k=i-1, k \neq i+2}^{i+2} \frac{x - x_{k-1/2}}{x_{i+3/2} - x_{k-1/2}}$$

We note $x = x_{i+1/2} - \alpha, \alpha > 0$ then

$$L_{i-1}(x) = \prod_{k=i-1, k \neq i-1}^{i+2} \frac{-\alpha + x_{i+1/2} - x_{k-1/2}}{x_{i-3/2} - x_{k-1/2}}$$

$$L_i(x) = \prod_{k=i-1, k \neq i}^{i+2} \frac{-\alpha + x_{i+1/2} - x_{k-1/2}}{x_{i-1/2} - x_{k-1/2}}$$

$$L_{i+1}(x) = \prod_{k=i-1, k \neq i+1}^{i+2} \frac{-\alpha + x_{i+1/2} - x_{k-1/2}}{x_{i+1/2} - x_{k-1/2}}$$

$$L_{i+2}(x) = \prod_{k=i-1, k \neq i+2}^{i+2} \frac{-\alpha + x_{i+1/2} - x_{k-1/2}}{x_{i+3/2} - x_{k-1/2}}$$

Assuming $\Delta x = x_{i+1/2} - x_{i-1/2}, \forall i$,

$$\begin{aligned} L_{i-1}(x) &= \prod_{k=i-1, k \neq i-1}^{i+2} \frac{-\alpha + \Delta x(i - k + 1)}{\Delta x(i - 1 - k)} \\ L_i(x) &= \prod_{k=i-1, k \neq i}^{i+2} \frac{-\alpha + \Delta x(i - k + 1)}{\Delta x(i - k)} \\ L_{i+1}(x) &= \prod_{k=i-1, k \neq i+1}^{i+2} \frac{-\alpha + \Delta x(i - k + 1)}{\Delta x(i + 1 - k)} \\ dL_{i-1}(x) &= \prod_{k=i-1, k \neq i+2}^{i+2} \frac{-\alpha + \Delta x(i - k + 1)}{\Delta x(i + 2 - k)} \end{aligned}$$

Writing $\beta = \alpha/\Delta x$, we deduce:

$$\begin{aligned} L_{i-1}(x) &= \prod_{k=i-1, k \neq i-1}^{i+2} \frac{-\beta + (i - k + 1)}{(i - 1 - k)} \\ L_i(x) &= \prod_{k=i-1, k \neq i}^{i+2} \frac{-\beta + (i - k + 1)}{(i - k)} \\ L_{i+1}(x) &= \prod_{k=i-1, k \neq i+1}^{i+2} \frac{-\beta + (i - k + 1)}{(i + 1 - k)} \\ L_{i+2}(x) &= \prod_{k=i-1, k \neq i+2}^{i+2} \frac{-\beta + (i - k + 1)}{(i + 2 - k)} \end{aligned}$$

and then,

$$\begin{aligned} L_{i-1}(x) &= \frac{-\beta + (i - (i) + 1)}{(i - 1 - (i))} \frac{-\beta + (i - (i + 1) + 1)}{i - 1 - (i + 1)} \frac{-\beta + (i - (i + 2) + 1)}{(i - 1 - (i + 2))} \\ L_i(x) &= \frac{-\beta + (i - (i - 1) + 1)}{(i - (i - 1))} \frac{-\beta + (i - (i + 1) + 1)}{(i - (i + 1))} \frac{-\beta + (i - (i + 2) + 1)}{(i - (i + 2))} \\ L_{i+1}(x) &= \frac{-\beta + (i - (i - 1) + 1)}{(i + 1 - (i - 1))} \frac{-\beta + (i - (i + 2) + 1)}{(i + 1 - (i + 2))} \frac{-\beta + (i - (i) + 1)}{(i + 1 - (i))} \\ L_{i+2}(x) &= \frac{-\beta + (i - (i - 1) + 1)}{(i + 2 - (i - 1))} \frac{-\beta + (i - (i) + 1)}{(i + 2 - (i))} \frac{-\beta + (i - (i + 1) + 1)}{(i + 2 - (i + 1))} \end{aligned}$$

$$\begin{aligned} L_{i-1}(x) &= \frac{-\beta + 1}{-1} \frac{-\beta}{-2} \frac{-\beta - 1}{-3} \\ L_i(x) &= \frac{-\beta + 2}{1} \frac{-\beta}{-1} \frac{-\beta - 1}{-2} \\ L_{i+1}(x) &= \frac{-\beta + 2}{2} \frac{-\beta - 1}{-1} \frac{-\beta + 1}{1} \\ L_{i+2}(x) &= \frac{-\beta + 2}{3} \frac{-\beta + 1}{2} \frac{-\beta}{1} \end{aligned}$$

$$\begin{aligned}
L_{i-1}(x) &= 1/6(\beta - 1)(\beta)(\beta + 1) \\
L_i(x) &= 1/2(\beta - 2)(\beta)(\beta + 1) \\
L_{i+1}(x) &= 1/2(\beta - 2)(\beta + 1)(\beta - 1) \\
L_{i+2}(x) &= -1/6(\beta - 2)(\beta - 1)(\beta)
\end{aligned}$$

$$\begin{aligned}
L_{i-1}(x) &= 1/6(\beta^3 - \beta) \\
L_i(x) &= -1/2(\beta^3 - \beta^2 - 2\beta) \\
L_{i+1}(x) &= 1/2(\beta^3 - 2\beta^2 - \beta + 2) \\
L_{i+2}(x) &= -1/6(\beta^3 - 3\beta^2 + 2\beta)
\end{aligned}$$

Hence

$$\begin{aligned}
G(x) &= 1/6G_{i-3/2}(\beta^3 - \beta) \\
&- 1/2G_{i-1/2}(\beta^3 - \beta^2 - 2\beta) \\
&+ 1/2G_{i+1/2}(\beta^3 - 2\beta^2 - \beta + 2) \\
&- 1/6G_{i+3/2}(\beta^3 - 3\beta^2 + 2\beta)
\end{aligned}$$

Finally, the interpolation can be written:

$$\begin{aligned}
G(x) &= \beta^3(1/6G_{i-3/2} - 1/2G_{i-1/2} + 1/2G_{i+1/2} - 1/6G_{i+3/2}) \\
&+ \beta^2(1/2G_{i-1/2} - G_{i+1/2} + 1/2G_{i+3/2}) \\
&+ \beta(-1/6G_{i-3/2} + G_{i-1/2} - 1/2G_{i+1/2} - 1/3G_{i+3/2}) \\
&+ G_{i+1/2}
\end{aligned}$$

We note $x = x_{i+1/2} - \Delta x\beta = x_{i-1/2} - \Delta x(\beta - 1)$
and by defining $\theta = 1 - \beta > 0$, $x = x_{i-1/2} + \Delta x\theta$, then

$$\begin{aligned}
G(x) &= 1/6G_{i-3/2}((1 - \theta)^3 - (1 - \theta)) \\
&- 1/2G_{i-1/2}((1 - \theta)^3 - (1 - \theta)^2 - 2(1 - \theta)) \\
&+ 1/2G_{i+1/2}((1 - \theta)^3 - 2(1 - \theta)^2 - (1 - \theta) + 2) \\
&- 1/6G_{i+3/2}((1 - \theta)^3 - 3(1 - \theta)^2 + 2(1 - \theta))
\end{aligned}$$

$$\begin{aligned}
G(x) &= 1/6G_{i-3/2}(-\theta^3 + 3\theta^2 - 2\theta) \\
&- 1/2G_{i-1/2}(-\theta^3 + 2\theta^2 + \theta - 2) \\
&+ 1/2G_{i+1/2}(-\theta^3 + \theta^2 + 2\theta) \\
&- 1/6G_{i+3/2}(-\theta^3 + \theta)
\end{aligned}$$

$$\begin{aligned}
G(x) &= \theta^3(-1/6G_{i-3/2} + 1/2G_{i-1/2} - 1/2G_{i+1/2} + 1/6G_{i+3/2}) \\
&+ \theta^2(1/2G_{i-3/2} - G_{i-1/2} + 1/2G_{i+1/2}) \\
&+ \theta(-1/3G_{i-3/2} - 1/2G_{i-1/2} + G_{i+1/2} - 1/6G_{i+3/2}) \\
&+ G_{i-1/2}
\end{aligned}$$

Using the previous formula with $x = x_{i+1/2} - \Delta x\beta$, $\beta < 0$, $i \rightarrow i+1$, $\theta \rightarrow -\beta$, we obtain for negative displacement

$$\begin{aligned} G(x) &= \beta^3(1/6G_{i-1/2} - 1/2G_{i+1/2} + 1/2G_{i+3/2} - 1/6G_{i+5/2}) \\ &+ \beta^2(1/2G_{i-1/2} - G_{i+1/2} + 1/2G_{i+3/2}) \\ &+ \beta(1/3G_{i-1/2} + 1/2G_{i+1/2} - G_{i+3/2} + 1/6G_{i+5/2}) \\ &+ G_{i+1/2} \end{aligned}$$

We obtain the following results with $x = x_{i+1/2} - \Delta x\beta$:
if $\beta > 0$,

$$\begin{aligned} G(x) &= \beta^3(1/6G_{i-3/2} - 1/2G_{i-1/2} + 1/2G_{i+1/2} - 1/6G_{i+3/2}) \\ &+ \beta^2(1/2G_{i-1/2} - G_{i+1/2} + 1/2G_{i+3/2}) \\ &+ \beta(-1/6G_{i-3/2} + G_{i-1/2} - 1/2G_{i+1/2} - 1/3G_{i+3/2}) \\ &+ G_{i+1/2} \end{aligned}$$

if $\beta < 0$,

$$\begin{aligned} G(x) &= \beta^3(1/6G_{i-1/2} - 1/2G_{i+1/2} + 1/2G_{i+3/2} - 1/6G_{i+5/2}) \\ &+ \beta^2(1/2G_{i-1/2} - G_{i+1/2} \\ &+ 1/2G_{i+3/2}) \\ &+ \beta(1/3G_{i-1/2} + 1/2G_{i+1/2} - G_{i+3/2} + 1/6G_{i+5/2}) \\ &+ G_{i+1/2} \end{aligned}$$

We define $g_i^n \Delta x = G_{i+1/2} - G_{i-1/2}$. The previous expressions become:
if $\beta > 0$,

$$\begin{aligned} G(x) &= \beta^3(1/6G_{i-3/2} - 1/6G_{i-1/2} - 2/6G_{i-1/2} \\ &+ 2/6G_{i+1/2} + 1/6G_{i+1/2} - 1/6G_{i+3/2}) \\ &+ \beta^2(1/2G_{i-1/2} - 1/2G_{i+1/2} - 1/2G_{i+1/2} + 1/2G_{i+3/2}) \\ &+ \beta(-1/6G_{i-3/2} + 1/6G_{i-1/2} + 5/6G_{i-1/2} \\ &- 5/6G_{i+1/2} + 1/3G_{i+1/2} - 1/3G_{i+3/2}) \\ &+ G_{i+1/2} \\ &= \beta^3(-1/6g_{i-1}^n + 1/3g_i^n - 1/6g_{i+1}^n) \\ &+ \beta^2(-1/2g_i^n + 1/2g_{i+1}^n) \\ &+ \beta(1/6g_{i-1}^n - 5/6g_i^n - 1/3g_{i+1}^n) \\ &+ G_{i+1/2} \end{aligned}$$

if $\beta < 0$,

$$\begin{aligned}
G(x) &= \beta^3(1/6G_{i-1/2} - 1/6G_{i+1/2} - 2/6G_{i+1/2}) \\
&+ 2/6G_{i+3/2} + 1/6G_{i+3/2} - 1/6G_{i+5/2}) \\
&+ \beta^2(1/2G_{i-1/2} - 1/2G_{i+1/2} - 1/2G_{i+1/2} + 1/2G_{i+3/2}) \\
&+ \beta(1/3G_{i-1/2} - 1/3G_{i+1/2} + 5/6G_{i+1/2} \\
&- 5/6G_{i+3/2} - 1/6G_{i+3/2} + 1/6G_{i+5/2}) \\
&+ G_{i+1/2} \\
&= \beta^3(-1/6g_i^n + 2/6g_{i+1}^n - 1/6g_{i+2}^n) \\
&+ \beta^2(-1/2g_i^n + 1/2g_{i+1}^n) \\
&+ \beta(-1/3g_i^n - 5/6g_{i+1}^n + 1/6g_{i+2}^n) \\
&+ G_{i+1/2}
\end{aligned}$$

Let now compare theses fluxe expressions with the regular expression we found for the Hermite formulation noticed $\phi_{k+1/2, j_{k+1/2}=k}(\beta)$.

7.2 Positive displacement: $\beta > 0, \delta = 1, j_{k+1/2} = k$

$$\begin{aligned}
\frac{\phi_{k+1/2, j_{k+1/2}=k}(\beta)}{\Delta x} &= g_{k-1/2}^+ (\beta^2(\beta - 1)) \\
&+ g_{k+1/2}^- (\beta(1 - \beta)^2) \\
&+ \bar{g}_k (\beta^2(3 - 2\beta)) \\
&= \beta^3(g_{k-1/2}^+ + g_{k+1/2}^- - 2\bar{g}_k) \\
&+ \beta^2(-g_{k-1/2}^+ - 2g_{k+1/2}^- + 3\bar{g}_k) \\
&+ \beta(g_{k+1/2}^-)
\end{aligned}$$

We have introduced the value of the distribution at the cell's faces that we note $g_{k-1/2}^+ = g(x \geq x_{k-1/2})$ and $g_{k+1/2}^- = g(x \leq x_{k+1/2})$. We set these coefficients:

$$\begin{cases} g_{k-1/2}^+ = 1/3g_{k-1}^n + 5/6g_k^n - 1/6g_{k+1}^n \\ g_{k+1/2}^- = -1/6g_{k-1}^n + 5/6g_k^n + 1/3g_{k+1}^n \end{cases}$$

that leads to:

$$\begin{aligned}
\frac{\phi_{k+1/2, j_{k+1/2}=k}(\beta)}{\Delta x} &= g_{k-1/2} (\beta^2(\beta - 1)) \\
&+ g_{k+1/2} (\beta(1 - \beta)^2) \\
&+ \bar{g}_k (\beta^2(3 - 2\beta)) \\
&= \beta^3(1/3g_{k-1}^n + 5/6g_k^n - 1/6g_{k+1}^n - 1/6g_{k-1}^n) \\
&+ 5/6g_k^n + 1/3g_{k+1}^n - 2g_k^n) \\
&+ \beta^2(-1/3g_{k-1}^n - 5/6g_k^n + 1/6g_{k+1}^n) \\
&+ 1/3g_{k-1}^n - 10/6g_k^n - 2/3g_{k+1}^n + 3g_{k+1}^n) \\
&+ \beta(-1/6g_{k-1}^n + 5/6g_k^n + 1/3g_{k+1}^n)
\end{aligned}$$

$$\begin{aligned}
\frac{\phi_{k+1/2, j_{k+1/2}=k}(\beta)}{\Delta x} &= g_{k-1/2} (\beta^2(\beta-1)) \\
&+ g_{k+1/2} (\beta(1-\beta)^2) \\
&+ \bar{g}_k (\beta^2(3-2\beta)) \\
&= \beta^3(1/6g_{k-1}^n - 1/3g_k^n + 1/6g_{k+1}^n) \\
&+ \beta^2(1/2g_k^n - 1/2g_{k+1}^n) \\
&+ \beta(-1/6g_{k-1}^n + 5/6g_k^n + 1/3g_{k+1}^n)
\end{aligned}$$

Remembering that

$$\begin{aligned}
G(\beta) &= \beta^3(-1/6g_{i-1}^n + 1/3g_i^n - 1/6g_{i+1}^n) \\
&+ \beta^2(-1/2g_i^n + 1/2g_{i+1}^n) \\
&+ \beta(1/6g_{i-1}^n - 5/6g_i^n + 1/3g_{i+1}^n) \\
&+ G_{i+1/2}
\end{aligned}$$

we deduce:

$$G(x) = -\phi_{k+1/2, j_{k+1/2}=k}(\beta) + G_{i+1/2}$$

7.3 Negative displacement $\delta = 0$

$$\begin{aligned}
\frac{\phi_{k+1/2, j_{k+1/2}=k+1}(\beta)}{\Delta x} &= g_{k+1/2} (\beta(\beta+1)^2) \\
&+ g_{k+3/2} (\beta^2(1+\beta)) \\
&+ g_{k+1}^n (\beta^2(-3-2\beta)) \\
&= \beta^3(g_{k+3/2}^- + g_{k+1/2}^+ - 2g_{k+1}^n) \\
&+ \beta^2(2g_{k+1/2}^+ + g_{k+3/2}^- - 3g_{k+1}^n) \\
&+ \beta(g_{k+1/2}^+)
\end{aligned}$$

$$\begin{aligned}
G(x) &= \beta^3(-1/6g_i^n + 2/6g_{i+1}^n - 1/6g_{i+2}^n) \\
&+ \beta^2(-1/2g_i^n + 1/2g_{i+1}^n) \\
&+ \beta(-1/3g_i^n - 5/6g_{i+1}^n + 1/6g_{i+2}^n) \\
&+ G_{i+1/2}
\end{aligned}$$

We set these coefficients:

$$\begin{cases} g_{k+1/2}^+ = 1/3g_k^n + 5/6g_{k+1}^n - 1/6g_{k+2}^n \\ g_{k+3/2}^- = -1/6g_k^n + 5/6g_{k+1}^n + 1/3g_{k+2}^n \end{cases}$$

$$\begin{aligned}
\frac{\phi_{k+1/2, j_{k+1/2}=k+1}(\beta)}{\Delta x} &= \beta^3(1/3g_k^n + 5/6g_{k+1}^n - 1/6g_{k+2}^n - 1/6g_k^n) \\
&+ 5/6g_{k+1}^n + 1/3g_{k+2}^n - 2g_{k+1}^n) \\
&+ \beta^2(2/3g_k^n + 10/6g_{k+1}^n - 2/6g_{k+2}^n - 1/6g_k^n) \\
&+ 5/6g_{k+1}^n + 1/3g_{k+2}^n - 3g_{k+1}^n) \\
&+ \beta(1/3g_k^n + 5/6g_{k+1}^n - 1/6g_{k+2}^n)
\end{aligned}$$

$$\begin{aligned}
\frac{\phi_{k+1/2,j_{k+1/2}=k+1}(\beta)}{\Delta x} &= \beta^3(1/6g_k^n - 1/3g_{k+1}^n + 1/6g_{k+2}^n) \\
&+ \beta^2(1/2g_k^n - 1/2g_{k+1}^n) \\
&+ \beta(1/3g_k^n + 5/6g_{k+1}^n - 1/6g_{k+2}^n)
\end{aligned}$$

Remembering that:

$$\begin{aligned}
G(x) &= \beta^3(-1/6g_i^n + 1/3g_{i+1}^n - 1/6g_{i+2}^n) \\
&+ \beta^2(-1/2g_i^n + 1/2g_{i+1}^n) \\
&+ \beta(-1/3g_i^n - 5/6g_{i+1}^n + 1/6g_{i+2}^n) \\
&+ G_{i+1/2}
\end{aligned}$$

we deduce

$$G(x) = -\phi_{k+1/2,j_{k+1/2}=k+1}(\beta)/\Delta x + G_{i+1/2}$$

To conclude,

$$\begin{aligned}
g_i^{new} &= -\phi_{k+1/2,j_{k+1/2}}(\beta)/\Delta x + G_{i+1/2} + G_{k-1/2,j_{k-1/2}}(\beta)/\Delta x - G_{i-1/2} \\
g_i^{new} &= g_i^n - (\phi_{k+1/2,j_{k+1/2}}(\beta) + \phi_{k-1/2,j_{k-1/2}}(\beta))/\Delta x
\end{aligned}$$

7.4 Annexe B: Algorithms

We present some algorithms which are representative of the Gysela algorithms. We change the notation of the flux $\phi_{i+1/2}$ to $H_{i+1/2}$ for practical reasons. The 1D advection equation are solved with Algo. (1).

Algorithm 1 Advection

Require: $(x_k)_{k=-1/2 \dots N+1/2}, (\bar{g}_k^n)_{k=0 \dots N}$

- 1: $a=1, b=-1, c=0, p=1, q=4$
- 2: $L, U = \text{MatrixCoefficient}(a, b, c, p, q)$
- 3: $(g_{k-1/2}^n)_{k=0 \dots N+1} = \text{CoefficientHermite}((\bar{g}_k^n)_{k=0 \dots N}, L, N, 0, N+1, a, b, c)$
- 4: $\beta = \text{FootCharacteristic}(\dots)$
- 5: **if** $\beta = 0$ **then**
- 6: $H_{0-1/2} = 0$
- 7: **else**
- 8: **if** $\beta > 0$ **then**
- 9: $H_{0-1/2} = \beta \bar{g}_0^n$
- 10: **else**
- 11: $H_{0-1/2} = \text{FluxComputation}(\bar{g}_0^n, g_{0-1/2}^n, g_{0+1/2}^n, \beta)$
- 12: **end if**
- 13: **end if**
- 14: **for** $i=1:N-1$ **do**
- 15: $\beta = \text{FootCharacteristic}()$
- 16: **if** $\beta = 0$ **then**
- 17: $H_{i+1/2} = 0$
- 18: **else**
- 19: **if** $\beta > 0$ **then**
- 20: $j=i$
- 21: **else**
- 22: $j=i+1$
- 23: **end if**
- 24: $H_{i+1/2} = \text{FluxComputation}(\bar{g}_j^n, g_{j-1/2}^n, g_{j+1/2}^n, \beta)$
- 25: **end if**
- 26: **end for**
- 27: $\beta = \text{FootCharacteristic}(\dots)$
- 28: **if** $\beta = 0$ **then**
- 29: $H_{N+1/2} = 0$
- 30: **else**
- 31: **if** $\beta < 0$ **then**
- 32: $H_{N+1/2} = \beta \bar{g}_N^n$
- 33: **else**
- 34: $H_{N+1/2} = \text{FluxComputation}(\bar{g}_N^n, g_{N-1/2}^n, g_{N+1/2}^n, \beta)$
- 35: **end if**
- 36: **end if**
- 37: **for** $i=1:N-1$ **do**
- 38: $g_i^{n+1} = g_i^n - (H_{i+1/2} - H_{(i-1)+1/2})$
- 39: **end for**
- 40: **return** $g_{k=0 \dots N}^{n+1}$

7.4.1 FluxComputation and FootCharacteristic

Algorithm 2 FluxComputation

Require: $\bar{g}_{j_k}^n, g_{j_k-1/2}^n, g_{j_k+1/2}^n, \beta$

- 1: **if** $\beta > 0$ **then**
- 2: $\delta = 0$
- 3: **else**
- 4: $\delta = 1$
- 5: **end if**
- 6: $H_{k+1/2} = g_{j_k-1/2} (\beta(1-\delta) + \beta^2(2-3\delta) + \beta^3)$
- 7: $H_{k+1/2} = H_{k+1/2} + g_{j_k+1/2} (\beta\delta + \beta^2(1-3\delta) + \beta^3)$
- 8: $H_{k+1/2} = H_{k+1/2} + \bar{g}_{j_k} (\beta^2(-3+6\delta) + \beta^3(-2))$
- 9: **return** $H_{k+1/2}$

Algorithm 3 FootCharacteristic

- 1: **return** $x_{i+1/2}^*$

7.4.2 Hermite coefficient computation for natural conditions

For a domain with natural boundary conditions, we compute the value of the distribution at the cell's faces $g_{k-1/2}$ and $g_{k+1/2}$. We want to solve: $AX = B$ and we decompose A as $A = LU$ with Algo. (5) then we solve $X = U^{-1}L^{-1}B$ with Algo. (4).

Algorithm 4 CoefficientHermiteNatural

Require: $(\bar{g}_k^n)_{k=0\dots N}, (L_k)_{k=0\dots N+1}, (U_k)_{k=0\dots N+1}, 0, N+1, a, b, c$

- 1: $X_0 = 5\bar{g}_0^n$
- 2: **for** $k = 1 : N$ **do**
- 3: $X_k = 3(\bar{g}_{k-1}^n + \bar{g}_k^n) - L_{k-1}X_{k-1}$
- 4: **end for**
- 5: $X_{N+1} = 5\bar{g}_N^n - L_{N+1}X_{N-1} - L_NX_N$
- 6: $X_{N+1} = \frac{X_{N+1}}{U_{N+1}}$
- 7: **for** $k = N : 2, -1$ **do**
- 8: $X_k = \frac{X_k - X_{k+1}}{U_k}$
- 9: **end for**
- 10: $X_1 = \frac{X_1 - (1-c/a)X_2}{U_1}$
- 11: $X_0 = \frac{X_0 - bX_1 - cX_2}{U_0}$
- 12: **return** $(g_{k-1/2})_{k=0\dots N+1}$

We compute the matrix coefficients of the LU decomposition.

$$[A] = [L] [U]$$

$$\begin{bmatrix}
a & b & c & 0 & \cdots & \cdots & 0 \\
p & q & p & 0 & \cdots & \cdots & \vdots \\
0 & p & q & p & 0 & \cdots & \vdots \\
\vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\
\vdots & \cdots & 0 & p & q & p & 0 \\
\vdots & \cdots & \cdots & 0 & p & q & p \\
0 & \cdots & \cdots & 0 & d & e & f
\end{bmatrix} =
\begin{bmatrix}
1 & 0 & \cdots & \cdots & \cdots & \cdots & 0 \\
L_1 & 1 & 0 & \cdots & \cdots & \cdots & \vdots \\
0 & L_2 & 1 & 0 & \cdots & \cdots & \vdots \\
\vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\
\vdots & \cdots & 0 & L_{N-2} & 1 & 0 & \vdots \\
\vdots & \cdots & \cdots & 0 & L_{N-1} & 1 & 0 \\
0 & \cdots & \cdots & 0 & L_{N+1} & L_N & 1
\end{bmatrix}
\begin{bmatrix}
U_0 & b & c & 0 & \cdots & \cdots & 0 \\
0 & U_1 & 1 - \frac{c}{a} & 0 & \cdots & \cdots & \vdots \\
\vdots & 0 & U_2 & 1 & 0 & \cdots & \vdots \\
\vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\
\vdots & \cdots & \cdots & 0 & U_{N-1} & 1 & 0 \\
\vdots & \cdots & \cdots & \cdots & 0 & U_N & 1 \\
0 & \cdots & \cdots & \cdots & \cdots & 0 & U_{N+1}
\end{bmatrix}$$

Algorithm 5 MatrixCoefficientNatural**Require:** a,b,c,d,e,f,p,q,0,N+1

```

1:  $U_0 = a$ 
2:  $L_0 = \frac{p}{U_0}$ 
3:  $U_1 = q - L_0 \left(1 - \frac{c}{a}\right)$ 
4:  $L_1 = \frac{p}{U_1}$ 
5:  $U_2 = q - L_1 \left(1 - \frac{c}{a}\right)$ 
6: for  $k = 2 : N - 1$  do
7:    $L_k = \frac{p}{U_k}$ 
8:    $U_{k+1} = q - L_k$ 
9: end for
10:  $L_{N+1} = \frac{d}{U_{N-1}}$ 
11:  $L_N = \frac{e - L_{N+1}}{U_N}$ 
12:  $d_{N+1} = f - U_N$ 
13: return  $L, U$ 

```

7.4.3 Hermite coefficient computation for periodic conditions

We compute the value of the distribution at the cell's faces $g_{k-1/2}$ and $g_{k+1/2}$. With periodic boundary conditions, we decompose A as $A = LDL^T$ with Algo. (7) then we solve $Y = D^{-1}L^{-1}B$ with Algo. (6).

Algorithm 6 CoefficientHermitePeriodic**Require:** $(\bar{g}_k^n)_{k=0 \dots N-1}, (L_k)_{k=0 \dots N}, (UD_k)_{k=0 \dots N}, a, b, c$

```

1:  $X_0 = 3(\bar{g}_N^n + \bar{g}_0^n)$ 
2: for  $k = 1 : N - 1$  do
3:    $X_k = 3(\bar{g}_k^n + \bar{g}_{k+1}^n) - L_{k-1}X_{k-1}$ 
4: end for
5:  $S = 0$ 
6: for  $m = 0 : N - 2$  do
7:    $S = S + \delta_m X_m$ 
8: end for
9:  $X_N = 3(\bar{g}_N^n + \bar{g}_0^n) - sum - L_{N-1}X_{N-1}$ 
10: for  $m = 0 : N$  do
11:    $X_m = \frac{X_m}{D_m}$ 
12: end for
13:  $X_{N-1} = X_{N-1} - L_{(N-1)}X_N$ 
14: for  $k = N - 2 : 0, -1$  do
15:    $X_k = X_k - L_k X_{k+1} - \delta_k X_N$ 
16: end for
17: return  $(g_{k-1/2})_{k=0 \dots N}$ 

```

We compute the matrix coefficients of the LDL decomposition.

$$\begin{aligned}
[A] &= [L] [D] [L^T] \\
\begin{bmatrix} q & p & 0 & 0 & \cdots & \cdots & 1 \\ p & q & p & 0 & \cdots & \cdots & \vdots \\ 0 & p & q & p & 0 & \cdots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \cdots & 0 & p & q & p & 0 \\ \vdots & \cdots & \cdots & 0 & p & q & p \\ 1 & \cdots & \cdots & 0 & 0 & p & q \end{bmatrix} &= \\
\begin{bmatrix} 1 & 0 & \cdots & \cdots & \cdots & \cdots & 0 \\ L_1 & 1 & 0 & \cdots & \cdots & \cdots & \vdots \\ 0 & L_2 & 1 & 0 & \cdots & \cdots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \cdots & 0 & L_{N-2} & 1 & 0 & \vdots \\ \vdots & \cdots & \cdots & 0 & L_{N-1} & 1 & 0 \\ \delta_1 & \delta_2 & \cdots & \cdots & \delta_{N-2} & L_{N-1} & 1 \end{bmatrix} \begin{bmatrix} D_1 & 0 & 0 & 0 & \cdots & \cdots & 0 \\ 0 & D_1 & 0 & 0 & \cdots & \cdots & \vdots \\ \vdots & 0 & D_2 & 0 & 0 & \cdots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \cdots & \cdots & 0 & D_{N-2} & 0 & 0 \\ \vdots & \cdots & \cdots & \cdots & 0 & D_{N-1} & 0 \\ 0 & \cdots & \cdots & \cdots & \cdots & 0 & D_N \end{bmatrix}
\end{aligned}$$

Algorithm 7 MatrixCoefficientPeriodic

Require: p, q, i_{start}, i_{end}

```

1:  $D_1 = q$ 
2:  $L_1 = \frac{p}{D_1}$ 
3:  $\delta_1 = \frac{1}{D_1}$ 
4: for  $k = 2 : N - 2$  do
5:    $D_k = q - pL_{k-1}$ 
6:    $L_k = \frac{p}{D_k}$ 
7:    $\delta_k = -\delta_{k-1} \frac{p}{D_k}$ 
8: end for
9:  $D_{N-1} = q - pL_{N-2}$ 
10:  $L_{N-1} = \frac{p - p\delta_{N-2}}{D_{N-1}}$ 
11:  $D_N = q - \sum_{j=1}^{N-2} D_j \delta_j^2 - D_{N-1} L_{N-1}^2$ 
12: return  $L, D$ 

```

References

- [1] J.-P. Braeunig, N. Crouseilles, V. Grandgirard, G. Latu, M. Mehrenberger, E. Sonnendrücker, Some numerical aspects of the conservative PSM scheme in a 4D drift-kinetic code, INRIA Research Report RR-7109 (november 2009).
- [2] A. J. Brizard and T. S. Hahm, Foundations of nonlinear gyrokinetic theory, *Rev. Mod. Phys.* 79, 421 (2007).
- [3] C.Z. Cheng, G. Knorr, The integration of the Vlasov equation in configuration space, *J. Comput. Phys.* 22, pp. 330-351 (1976).
- [4] N. Crouseilles, M. Mehrenberger, E. Sonnendrücker, Conservative semi-Lagrangian schemes for Vlasov equations, *J. Comput. Phys.*, 229, (2010), pp 1927-1953.
- [5] N. Crouseilles, G. Latu, E. Sonnendrücker, A Vlasov solver based on local cubic spline interpolation on patches, *J. Comput. Phys.*, Vol. 228, pp. 1429-1446 (2009).
- [6] A. M. Dimits et al, Comparisons and physics basis of tokamak transport models and turbulence simulations, *Phys. Plasmas*, 7, 969 (2000).
- [7] F. Filbet, E. Sonnendrücker and P. Bertrand, Conservative Numerical Schemes for the Vlasov Equation, *J. Comput. Physics* 172 (2001), no. 1, 166–187.
- [8] V. Grandgirard, Y. Sarazin, P. Angelino, A. Bottino, N. Crouseilles, G. Darmet, G. Dif-Pradalier, X. Garbet, Ph. Ghendrih, S. Jolliet, G. Latu, E. Sonnendrücker, L. Villard, Global full-f gyrokinetic simulations of plasma turbulence, *Plasma Phys. Control. Fus.*, Volume 49B, pp. 173–182 (december 2007).
- [9] V. Grandgirard, M. Brunetti, P. Bertrand, N. Besse, X. Garbet, P. Ghendrih, G. Manfredi, Y. Sarazin, O. Sauter, E. Sonnendrücker, J. Vaclavik, L. Villard, A drift-kinetic Semi-Lagrangian 4D code for ion turbulence simulation, *J. Comput. Physics*, vol. 217, no2, pp. 395-423 (2006).
- [10] F. Huot, A. Ghizzo, P. Bertrand, E. Sonnendrücker, O. Coulaud, Instability of the time splitting scheme for the one-dimensional and relativistic Vlasov-Maxwell system, *J. Comput. Phys.*, Vol. 185, Issue 2, pp. 512–531 (2003).
- [11] R.J. LeVeque, *Numerical Methods for Conservation Laws*, Birkhäuser (1990).
- [12] M. Shoucri, A two-level implicit scheme for the numerical solution of the linearized vorticity equation, *Int. J. Numer. Meth. Eng.* 17, p. 1525 (1981).
- [13] E. Sonnendrücker, J.R. Roche, P. Bertrand, A. Ghizzo, The Semi-Lagrangian Method for the Numerical Resolution of Vlasov Equations, *J. Comput. Physics*, Vol.149, No.2, pp. 201-220 (1999).

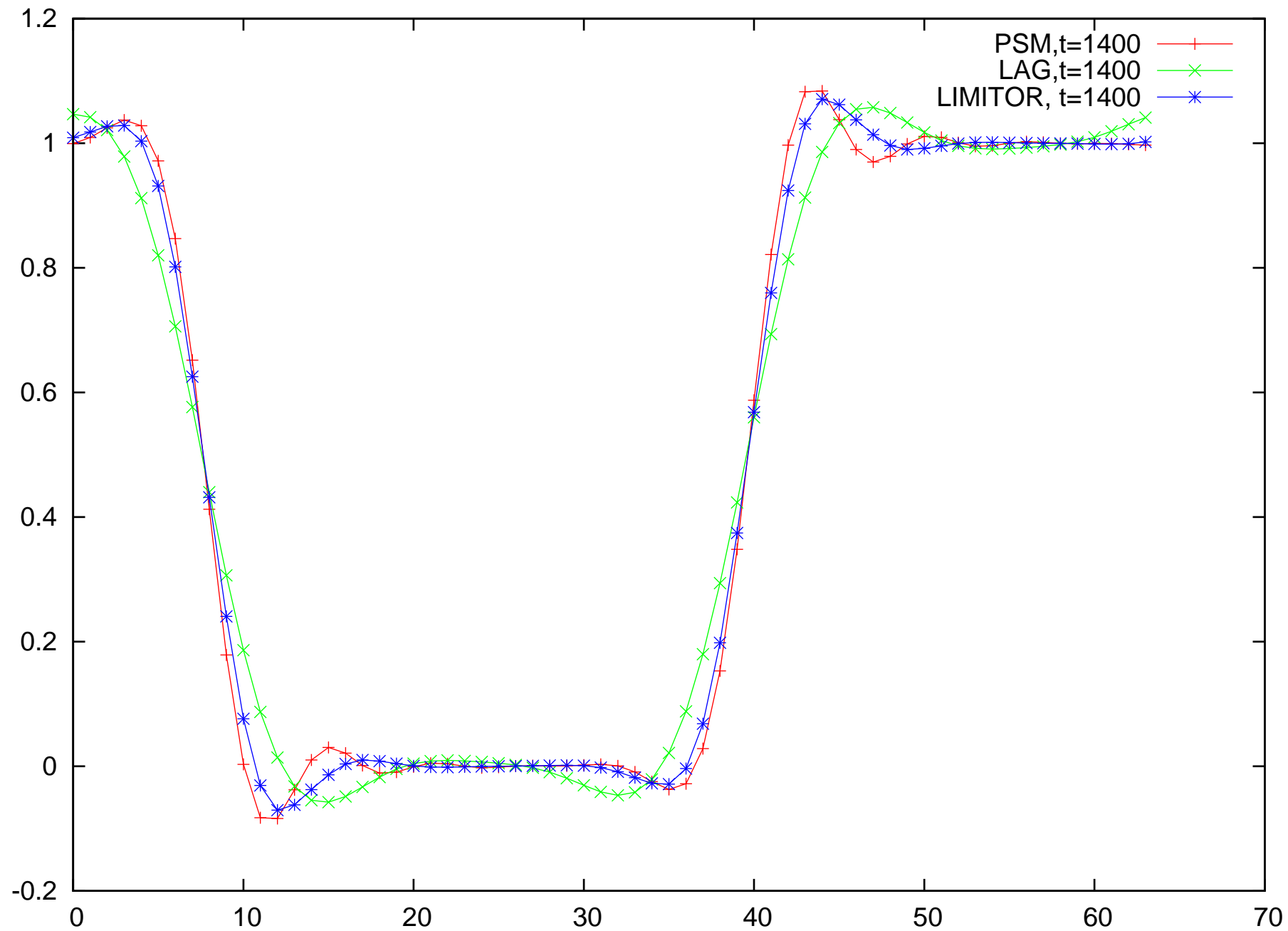
-
- [14] E. Sonnendrücker, Lecture notes CEA-EDF-INRIA, Modèles numériques pour la fusion contrôlée, Nice (Septembre 2008).
 - [15] T. Umeda, A conservative and non-oscillatory scheme for Vlasov code simulations, *Earth Planets Space*, 60, pp. 773-779, (2008).
 - [16] M. Zerroukat, N. Wood, A. Staniforth, The parabolic spline method (PSM) for conservative transport problems. *Int. J. Numer. Meth. Fluid.* v11. 1297-1318 (2006).
 - [17] M. Zerroukat, N. Wood, A. Staniforth, Application of the parabolic spline method (PSM) to a multi-dimensional conservative semi-Lagrangian transport scheme (SLICE), *J. Comput. Phys*, Vol. 225 n.1, pp. 935-948 (2007).

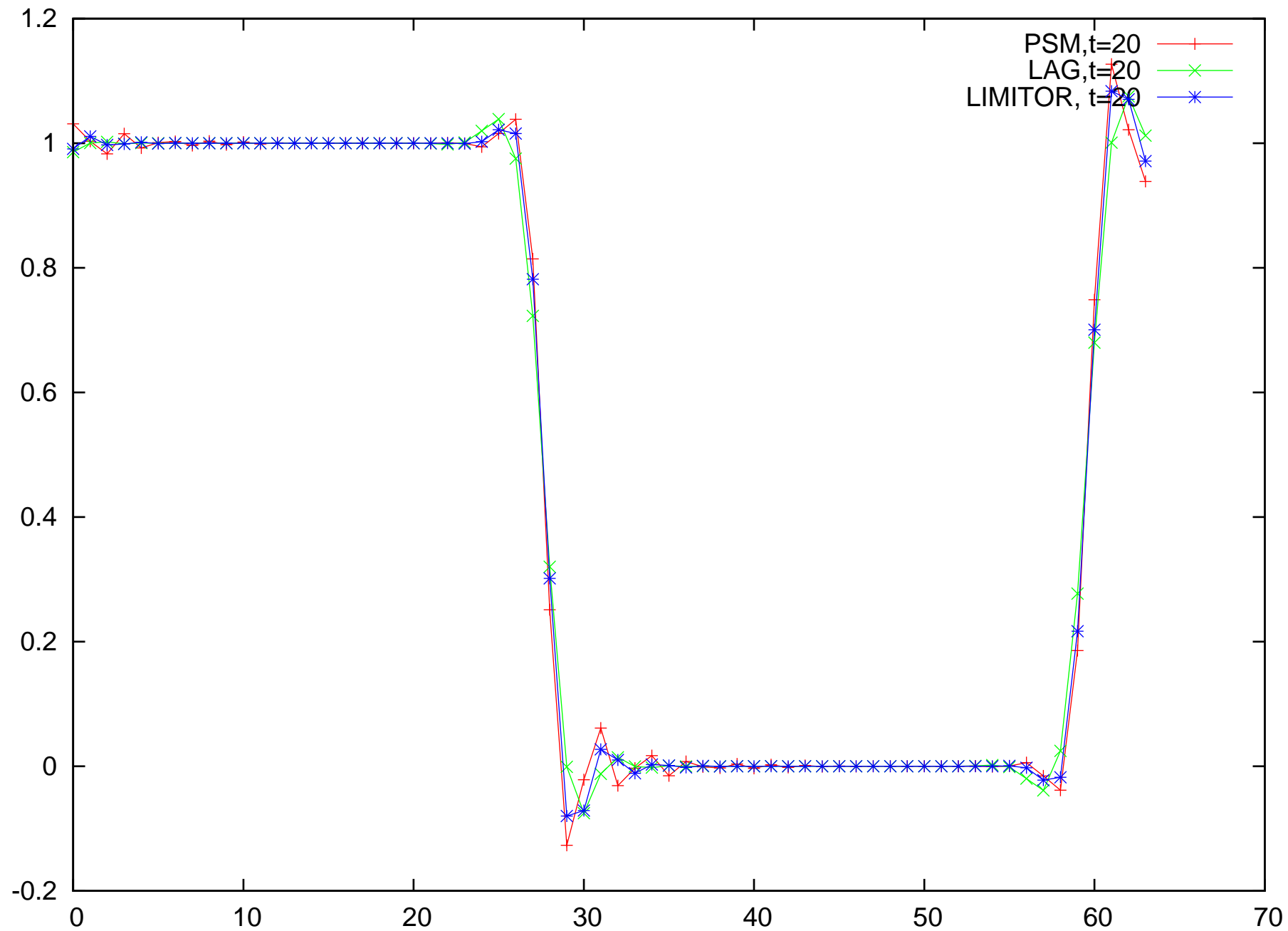


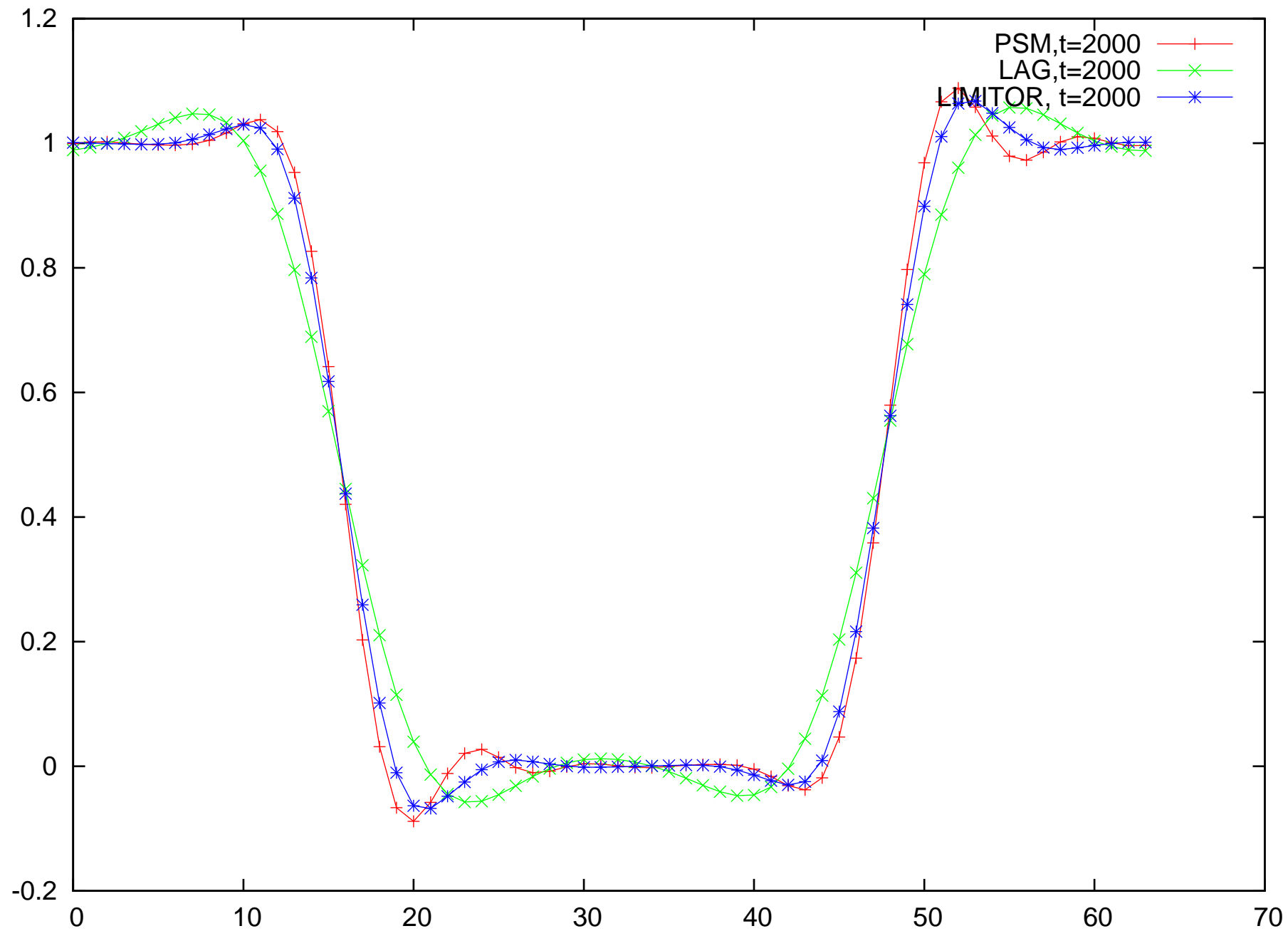
Centre de recherche INRIA Nancy – Grand Est
LORIA, Technopôle de Nancy-Brabois - Campus scientifique
615, rue du Jardin Botanique - BP 101 - 54602 Villers-lès-Nancy Cedex (France)

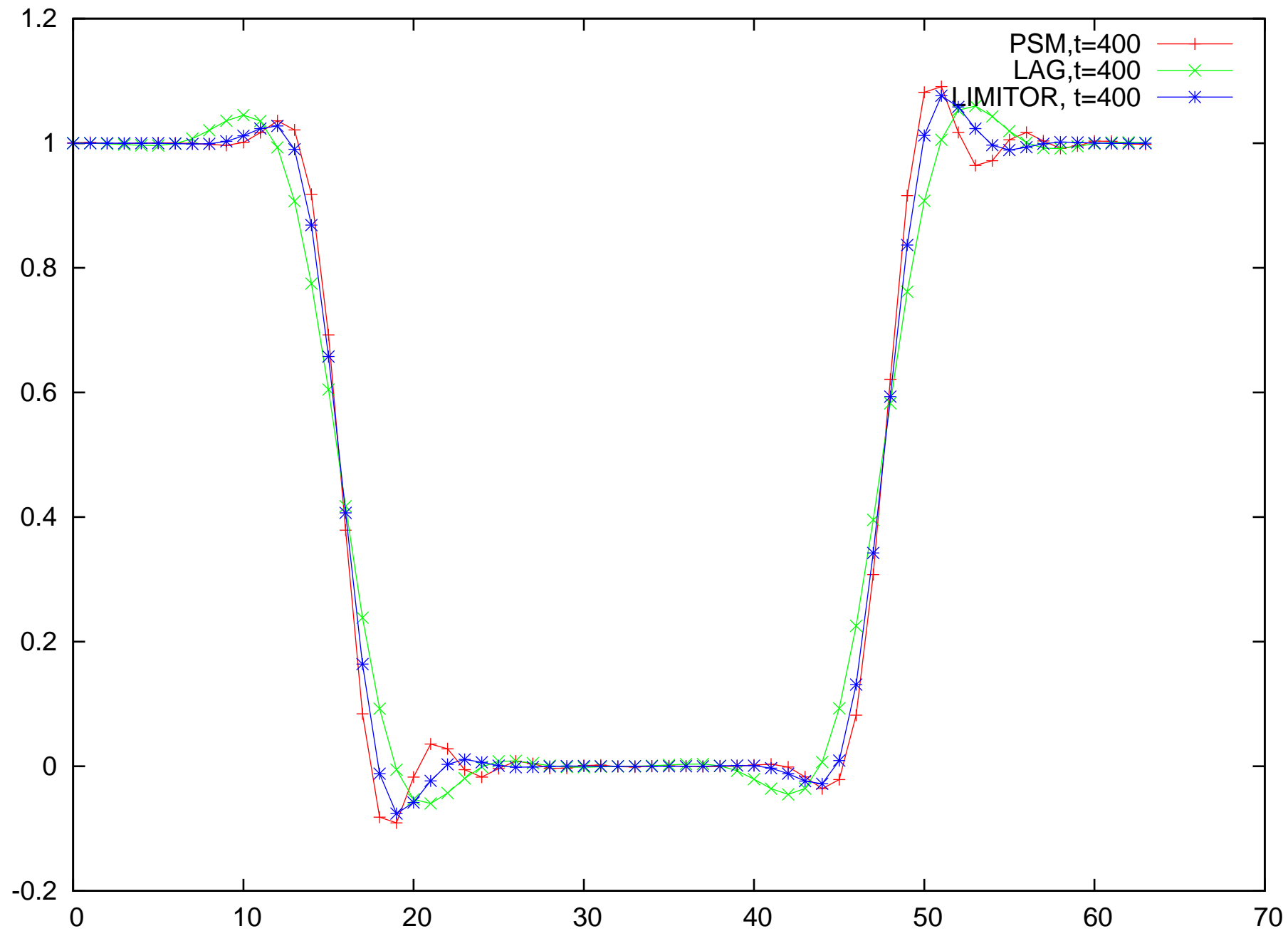
Centre de recherche INRIA Bordeaux – Sud Ouest : Domaine Universitaire - 351, cours de la Libération - 33405 Talence Cedex
Centre de recherche INRIA Grenoble – Rhône-Alpes : 655, avenue de l'Europe - 38334 Montbonnot Saint-Ismier
Centre de recherche INRIA Lille – Nord Europe : Parc Scientifique de la Haute Borne - 40, avenue Halley - 59650 Villeneuve d'Ascq
Centre de recherche INRIA Paris – Rocquencourt : Domaine de Voluceau - Rocquencourt - BP 105 - 78153 Le Chesnay Cedex
Centre de recherche INRIA Rennes – Bretagne Atlantique : IRISA, Campus universitaire de Beaulieu - 35042 Rennes Cedex
Centre de recherche INRIA Saclay – Île-de-France : Parc Orsay Université - ZAC des Vignes : 4, rue Jacques Monod - 91893 Orsay Cedex
Centre de recherche INRIA Sophia Antipolis – Méditerranée : 2004, route des Lucioles - BP 93 - 06902 Sophia Antipolis Cedex

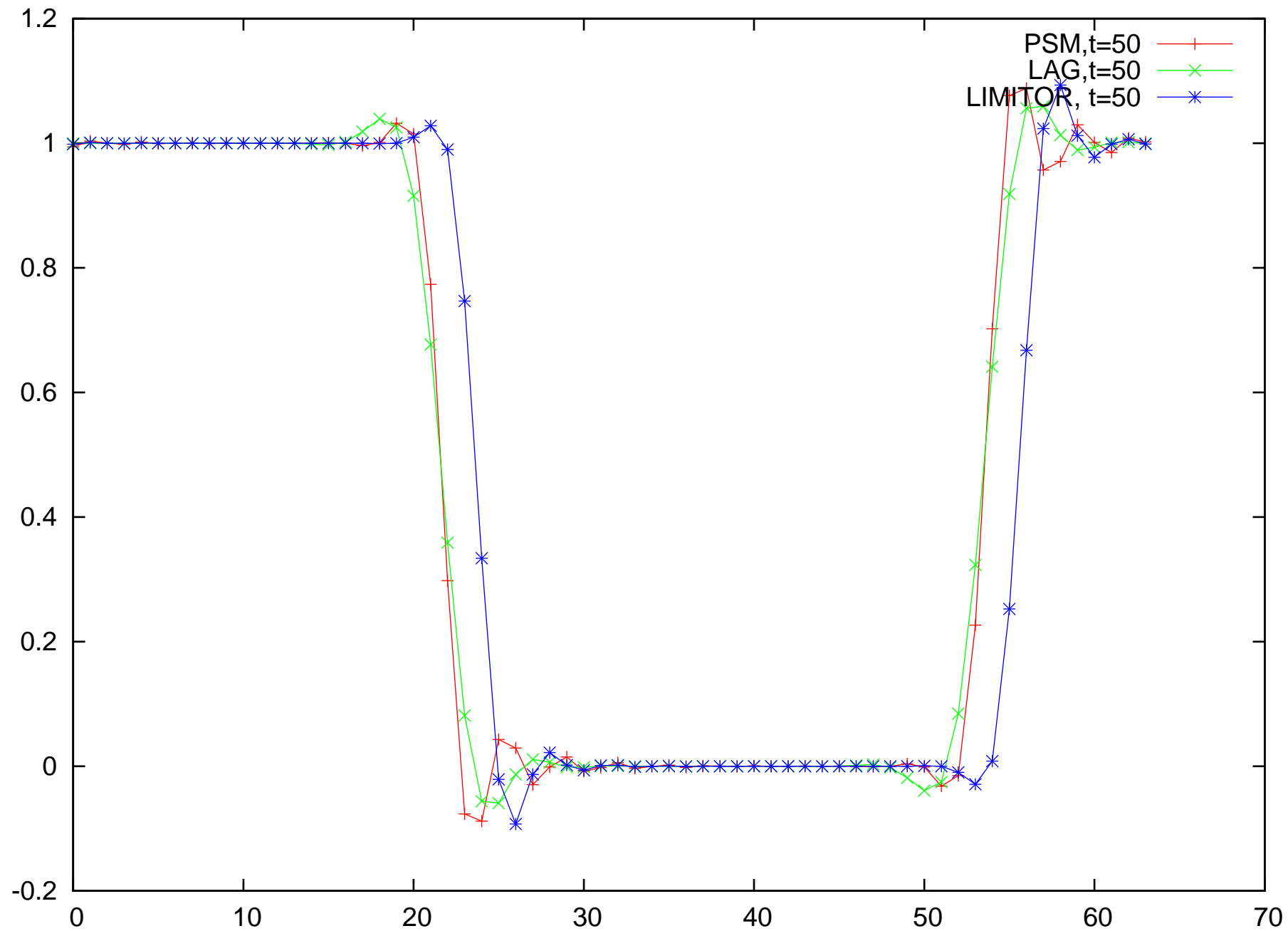
Éditeur
INRIA - Domaine de Voluceau - Rocquencourt, BP 105 - 78153 Le Chesnay Cedex (France)
<http://www.inria.fr>
ISSN 0249-6399

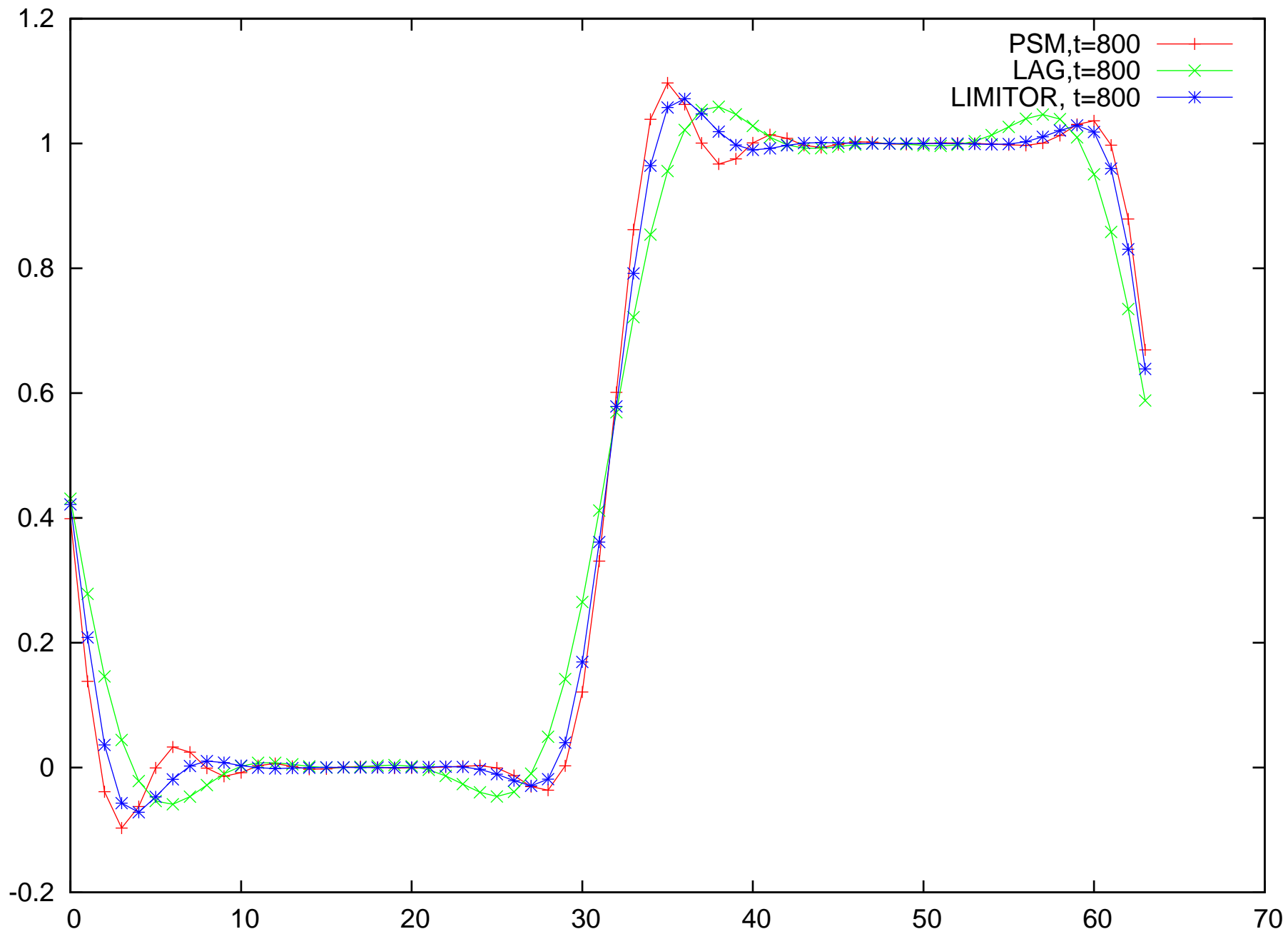




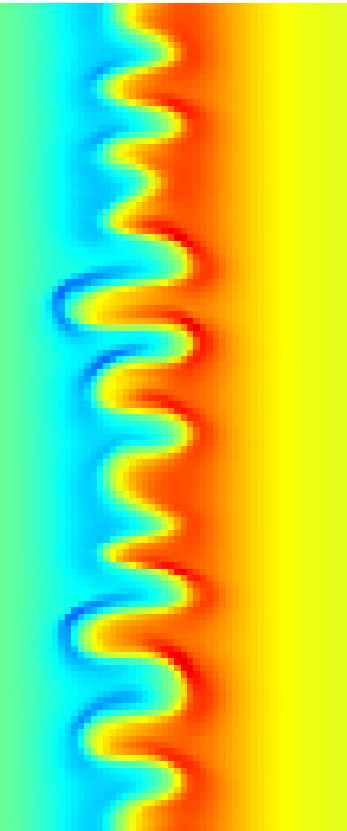






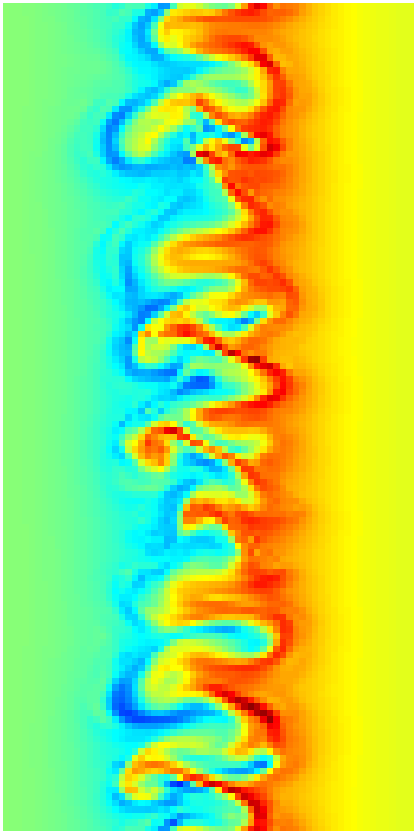


$C=0, t=750$



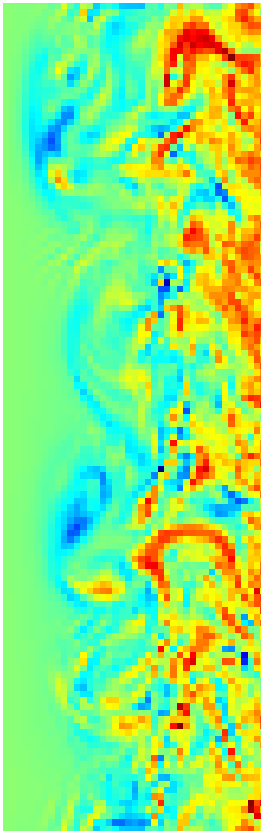
r

$C=0, t=1000$



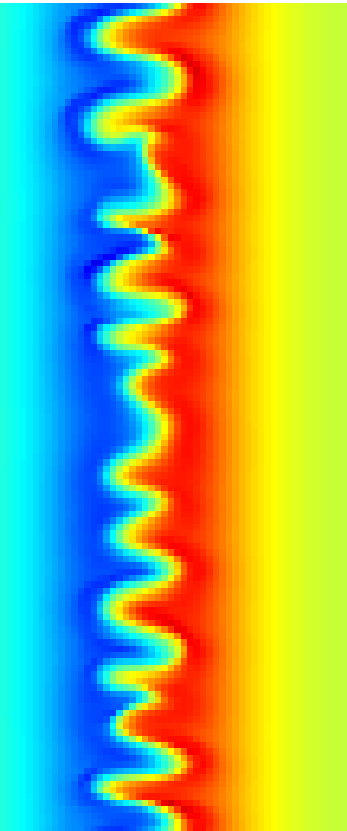
r

$C=0, t=1500$



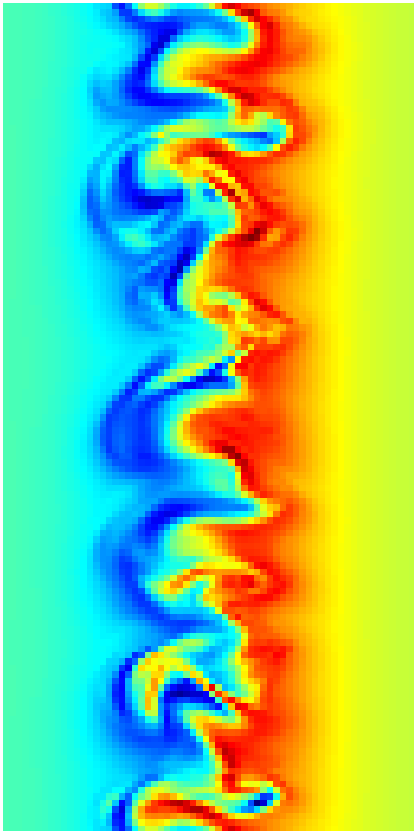
r

C=2,t=750



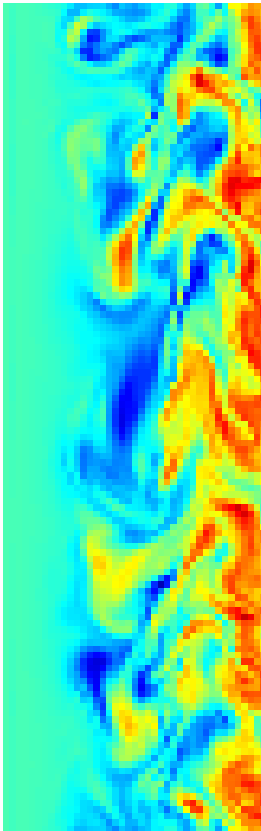
r

C=2,t=1000



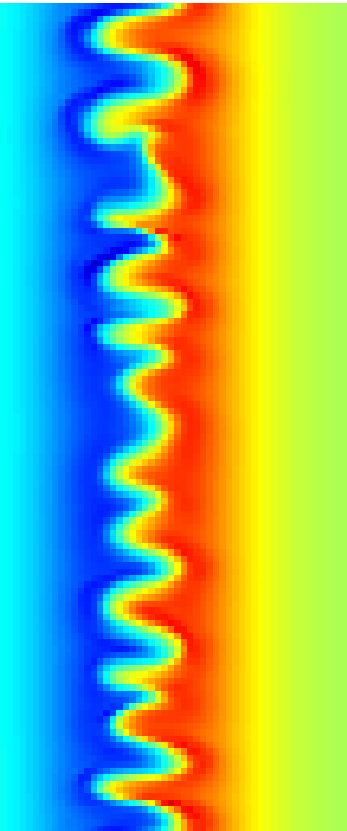
r

C=2,t=1500



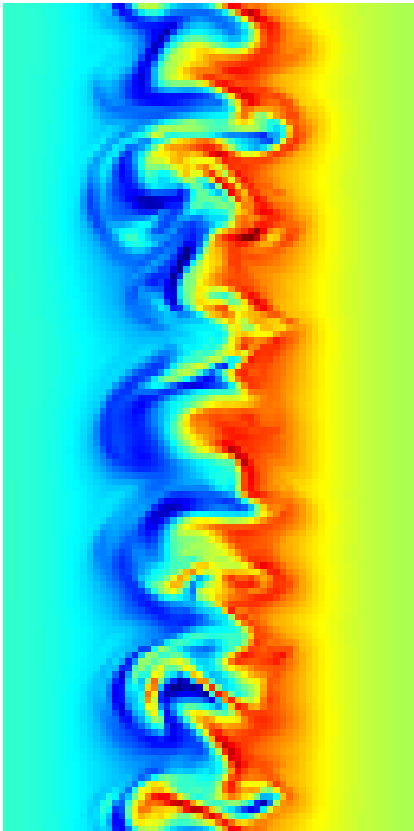
r

$C=3, t=750$



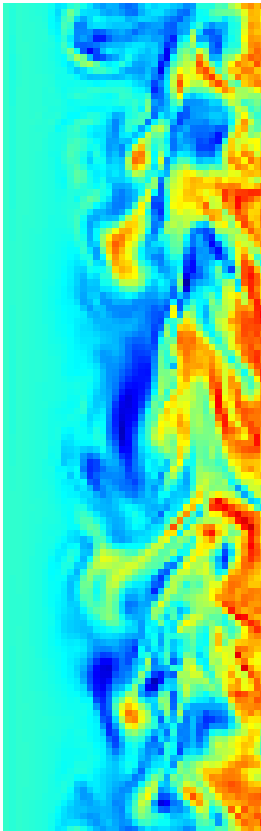
r

$C=3, t=1000$



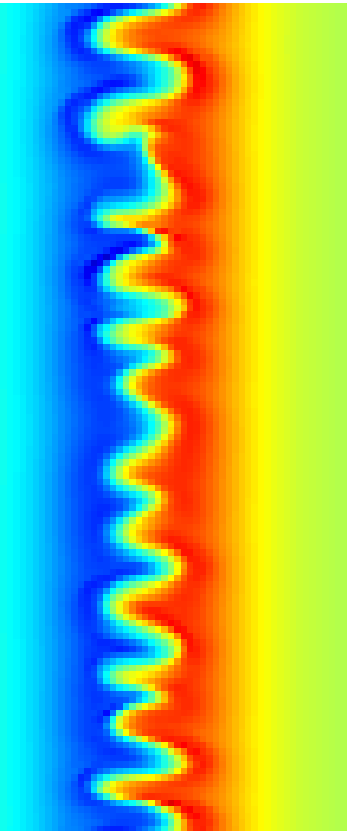
r

$C=3, t=1500$



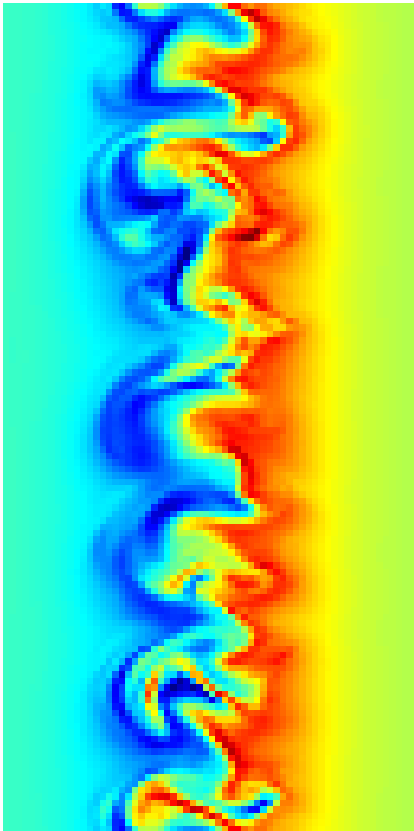
r

$C=4, t=750$



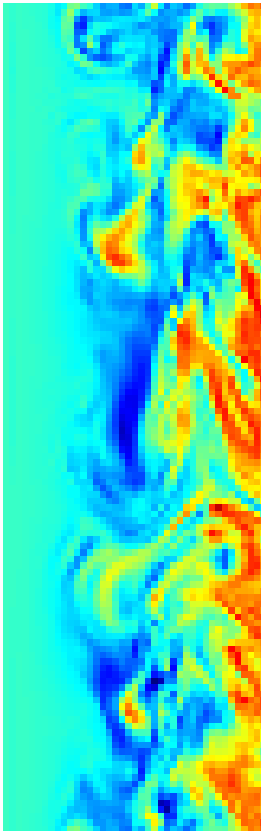
r

$C=4, t=1000$



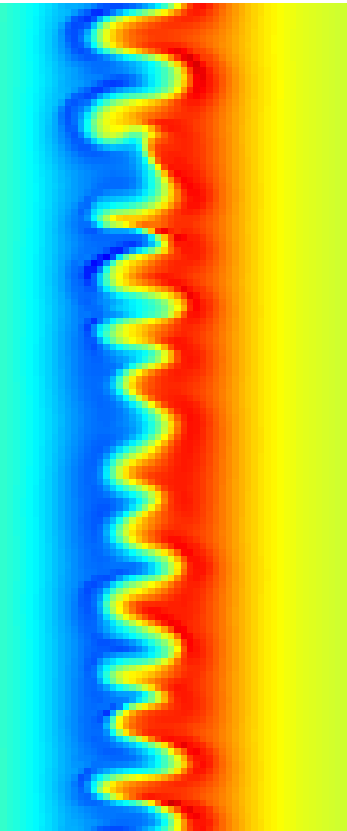
r

$C=4, t=1500$



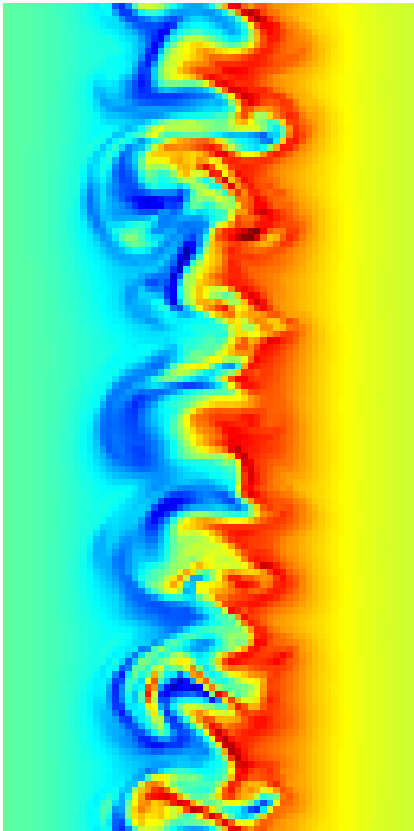
r

$C=5, t=750$



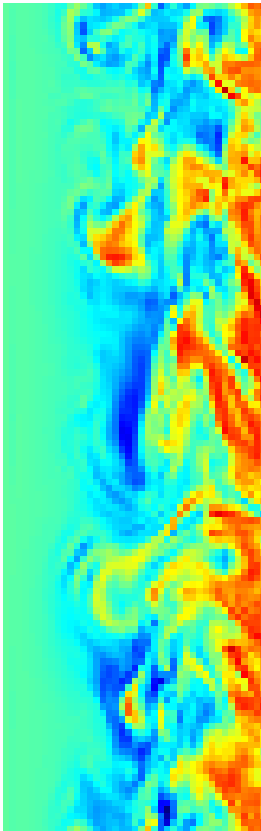
r

$C=5, t=1000$



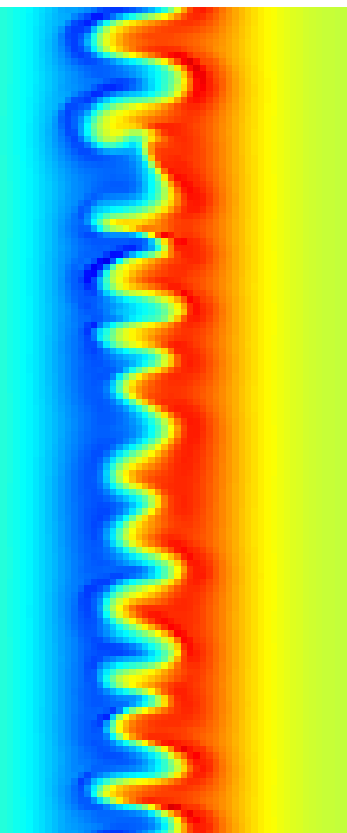
r

$C=5, t=1500$



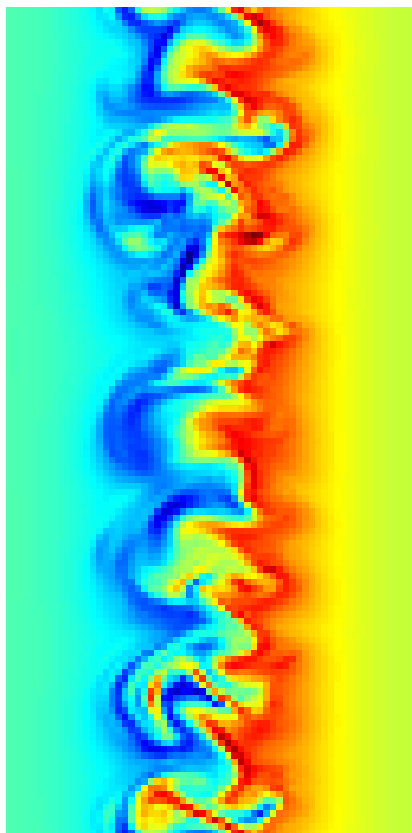
r

$C=10, t=750$



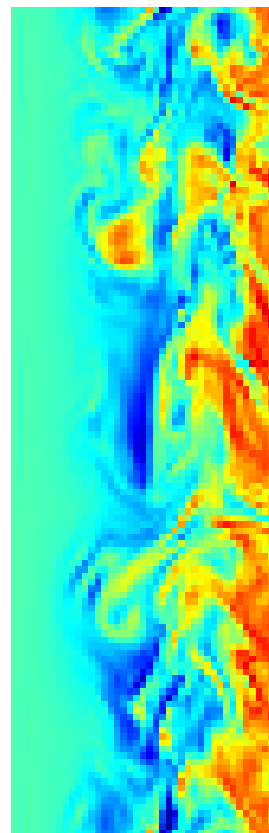
r

$C=10, t=1000$



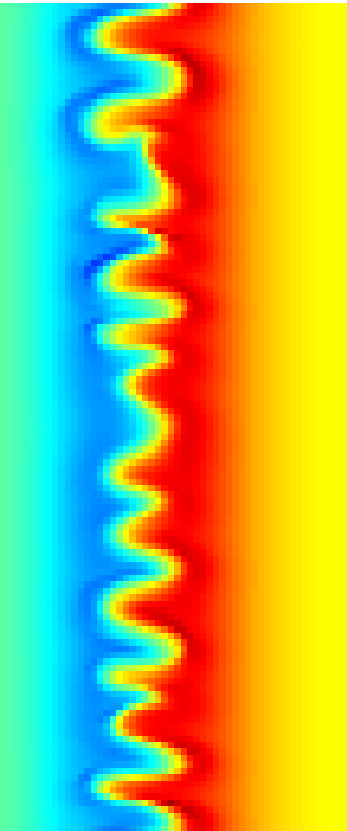
r

$C=10, t=1000$



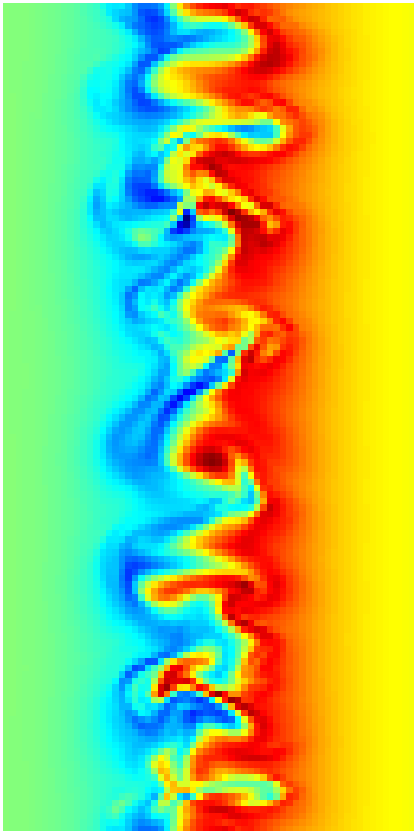
r

$C=1.5, t=750$



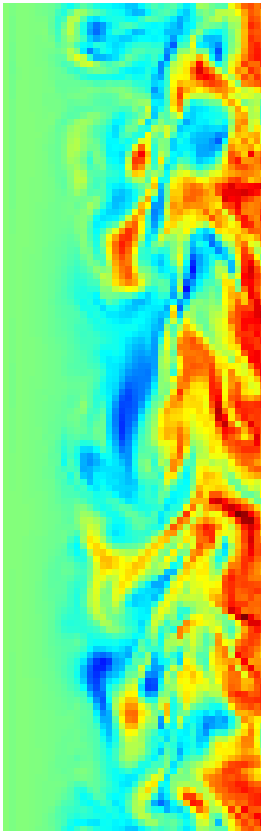
r

$C=1.5, t=1000$



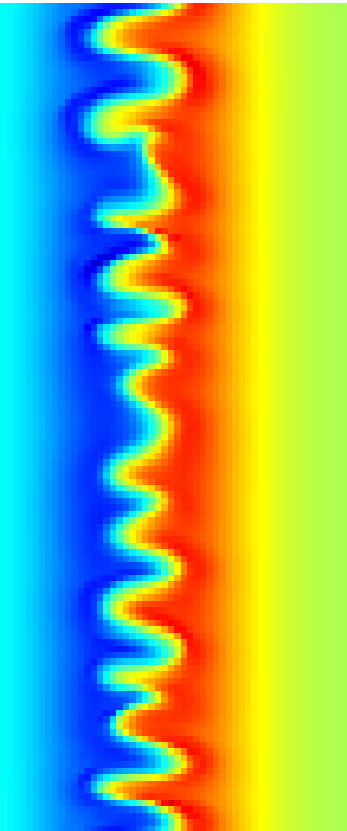
r

$C=1.5, t=1000$



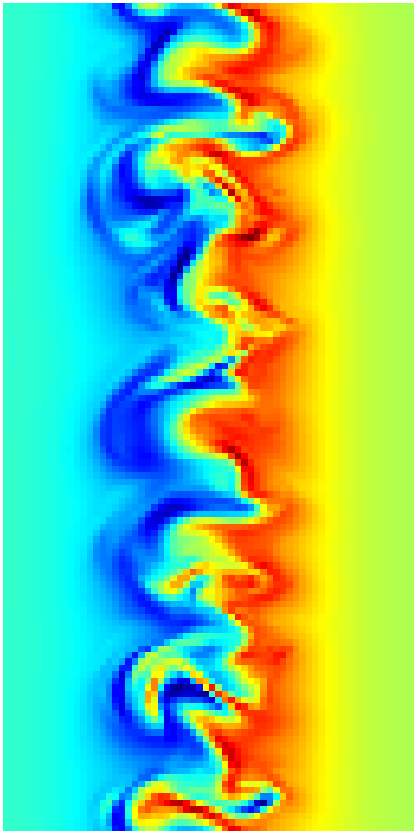
r

$C=2.5, t=750$



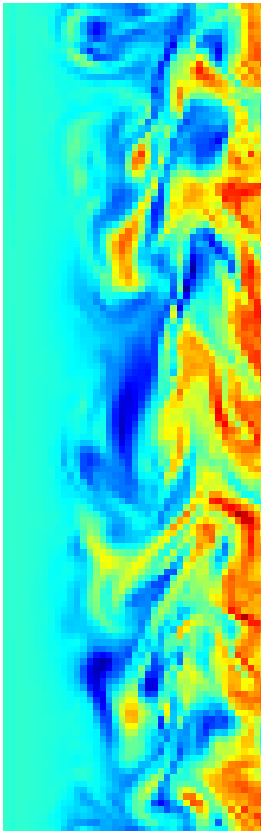
r

$C=2.5, t=1000$



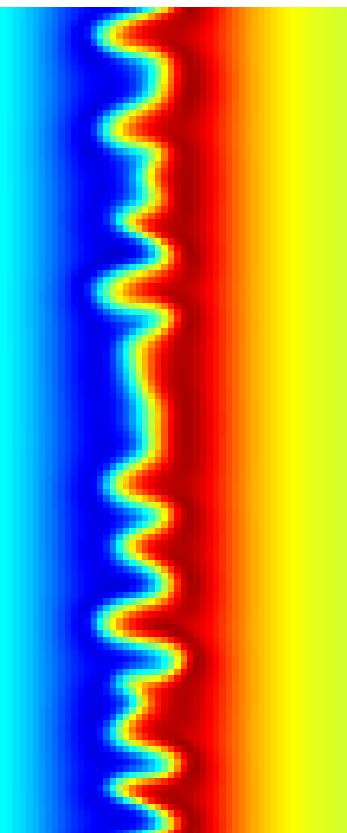
r

$C=2.5, t=1000$



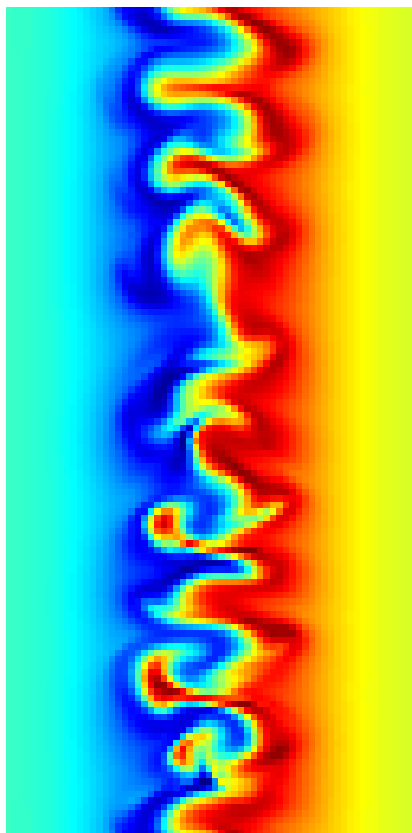
r

LAG, $t=750$



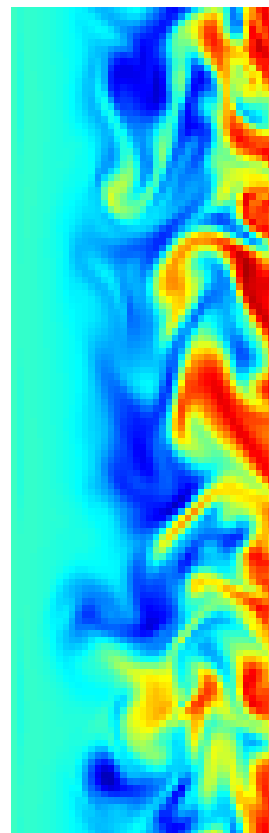
r

LAG, $t=1000$



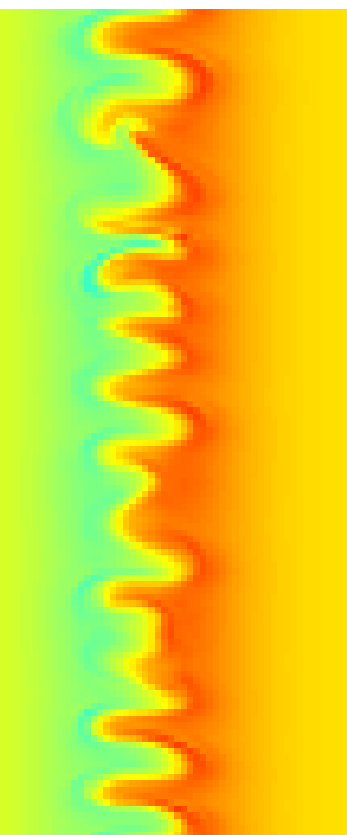
r

LAG, $t=1500$



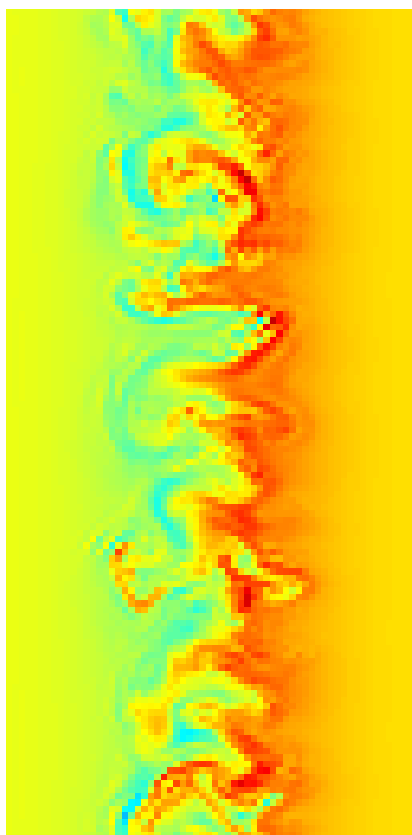
r

PSM,t=750



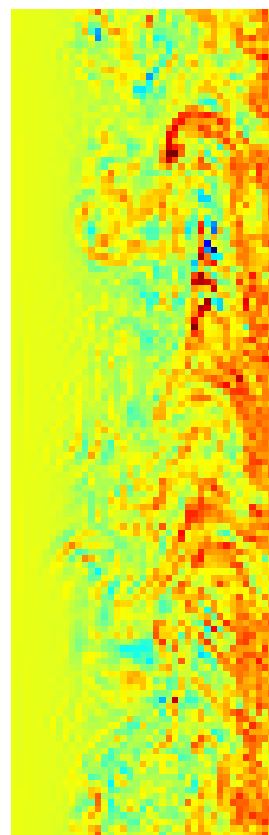
r

PSM,t=1000



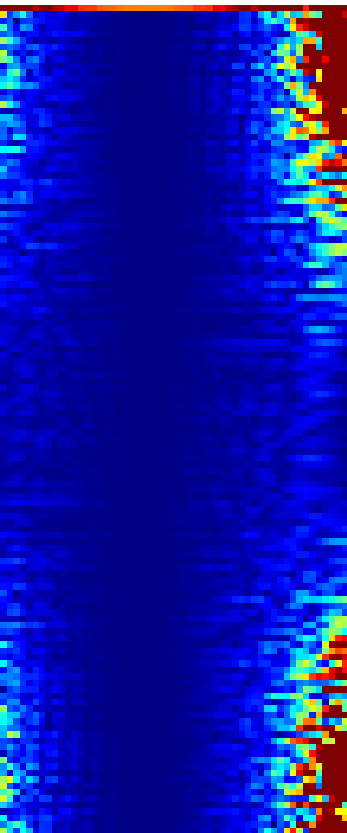
r

PSM,t=1500



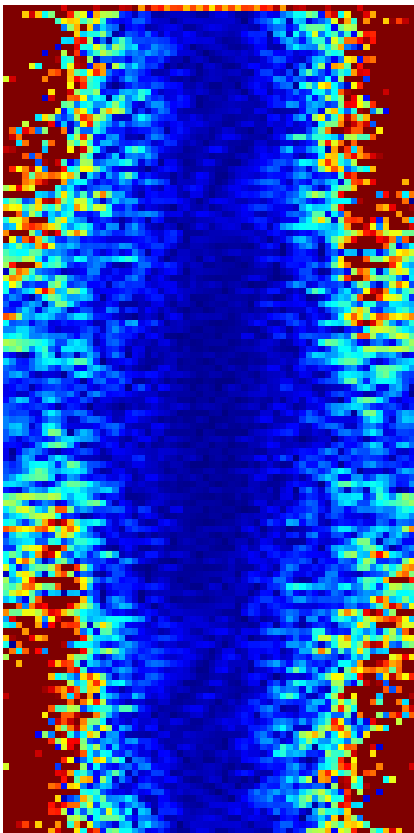
r

$C=0, t=750$



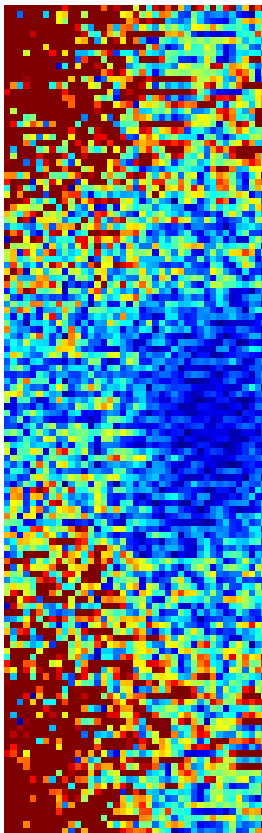
r

$C=0, t=1000$

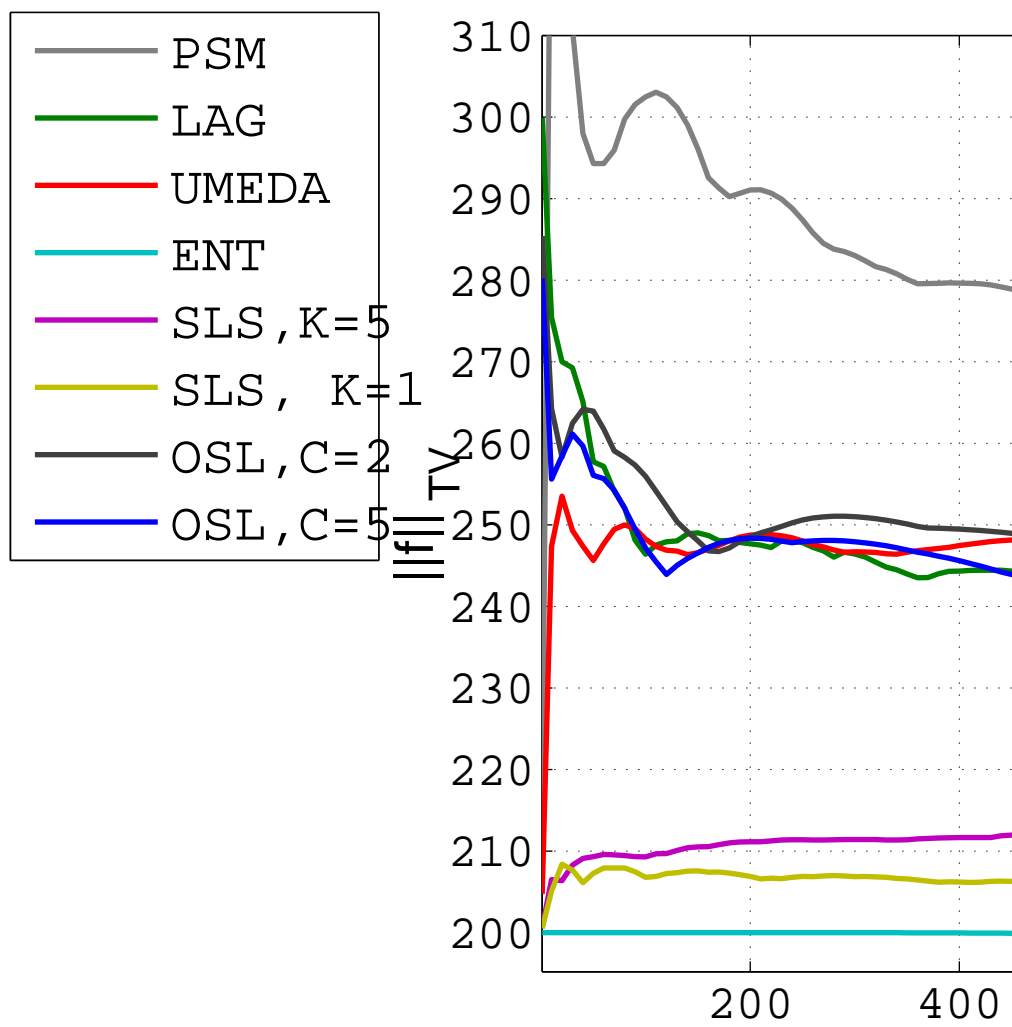
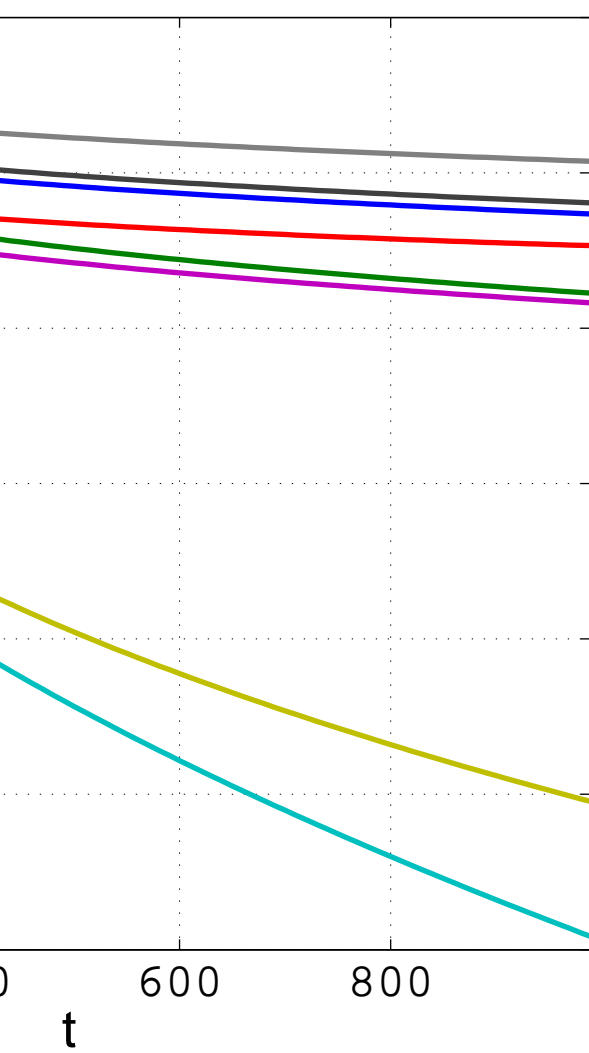


k_r

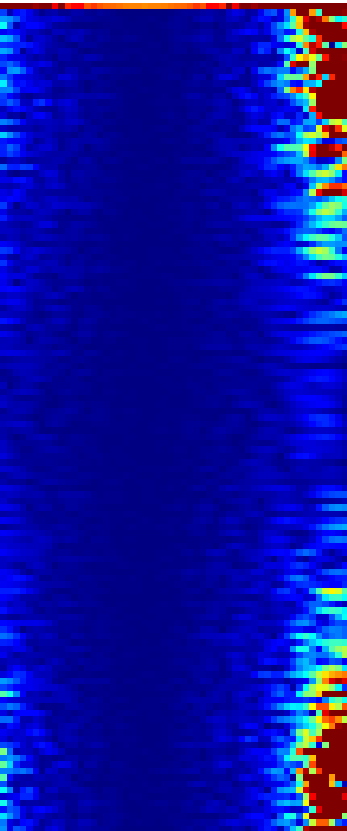
$C=0, t=1500$



k_r

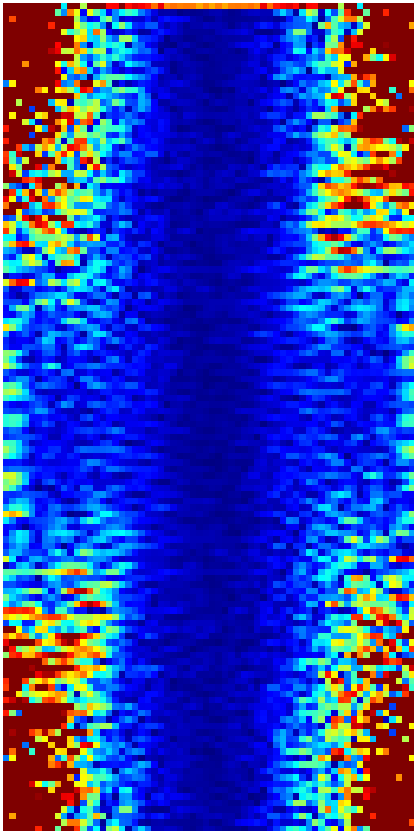


C=2,t=750



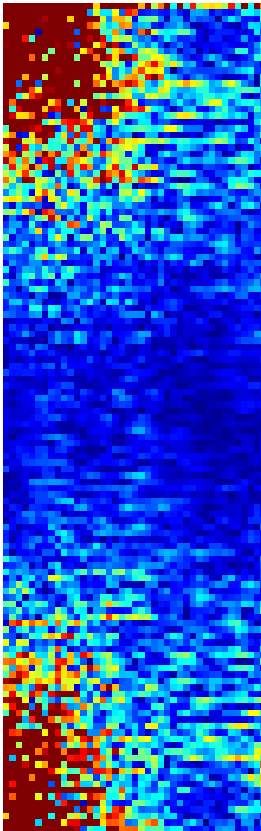
r

C=2,t=1000



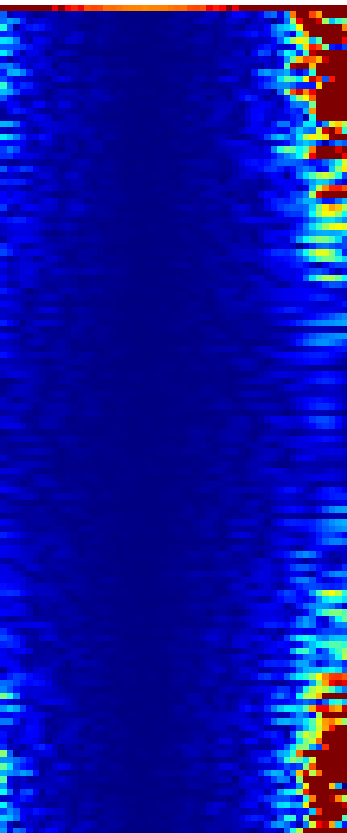
k_r

C=2,t=1500



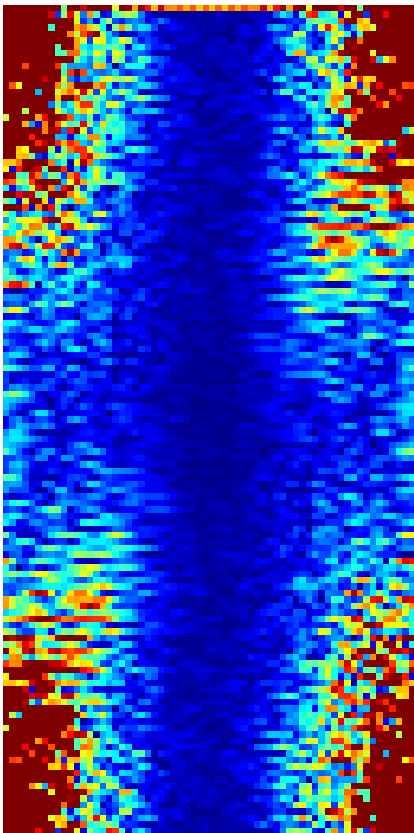
k_r

C=3,t=750



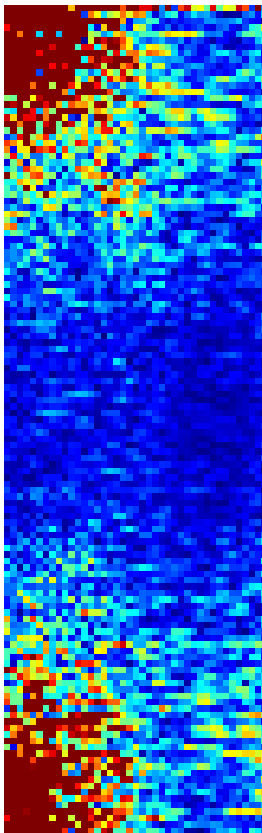
r

C=3,t=1000



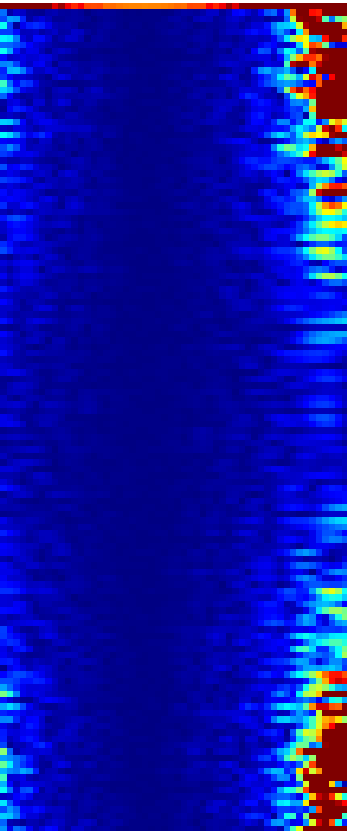
k_r

C=3,t=1500



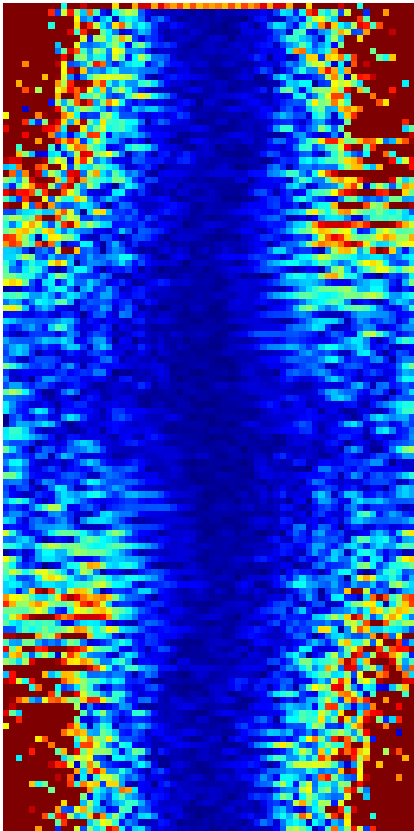
k_r

$C=4, t=750$



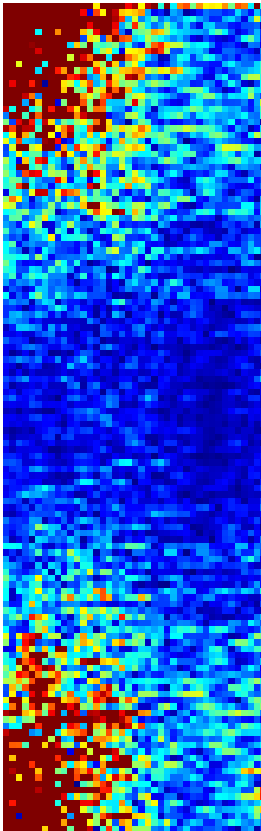
r

$C=4, t=1000$



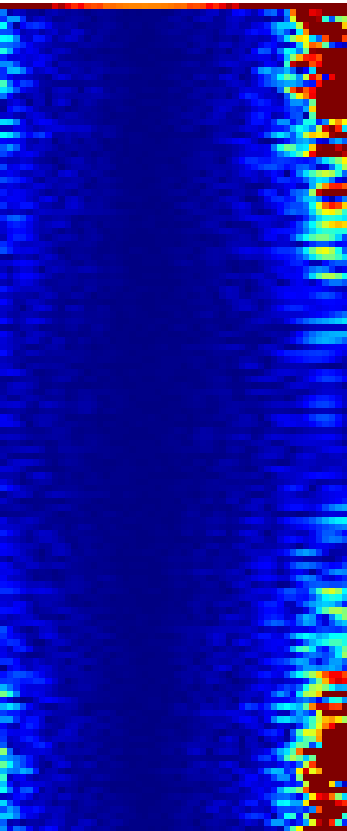
k_r

$C=4, t=1500$



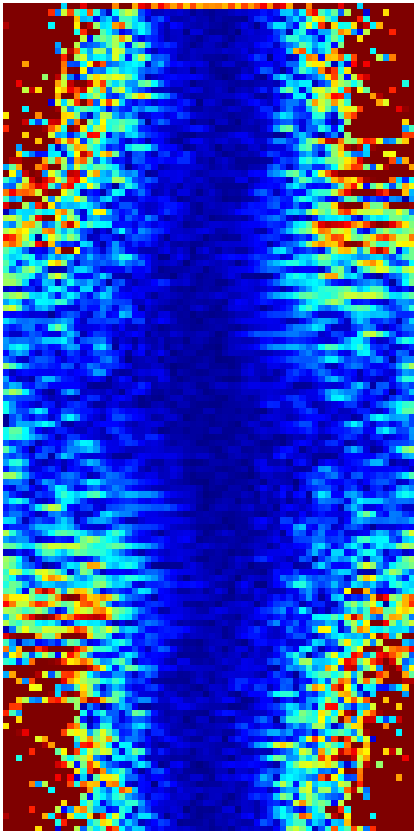
k_r

C=5,t=750



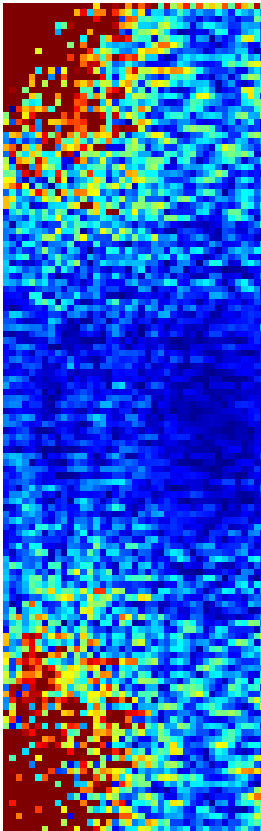
r

C=5,t=1000



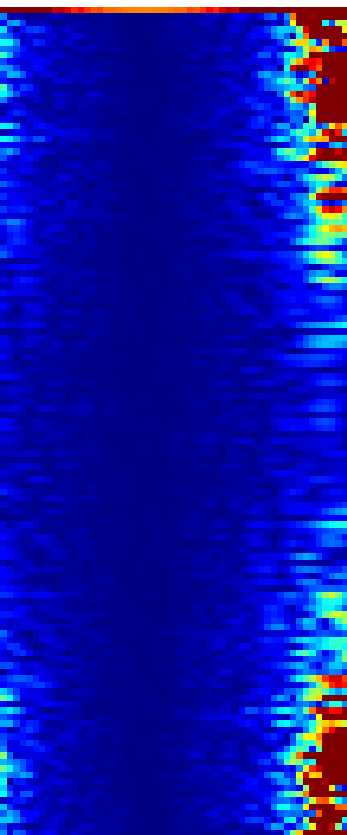
k_r

C=5,t=1500



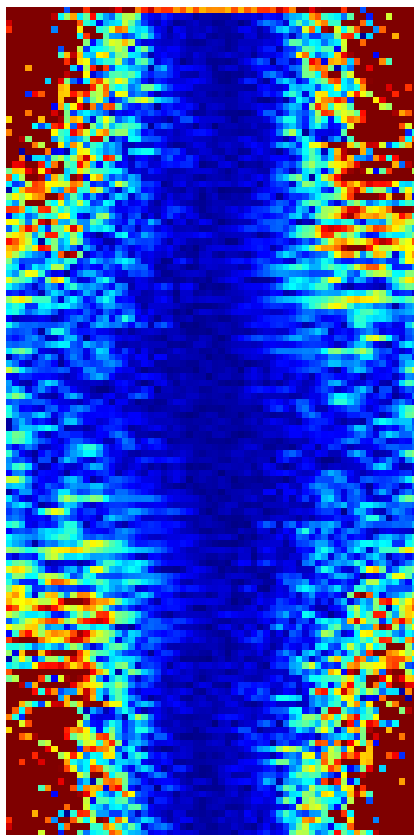
k_r

$C=10, t=750$



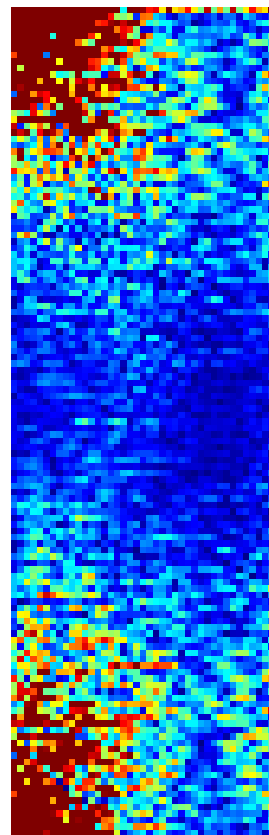
r

$C=10, t=1000$



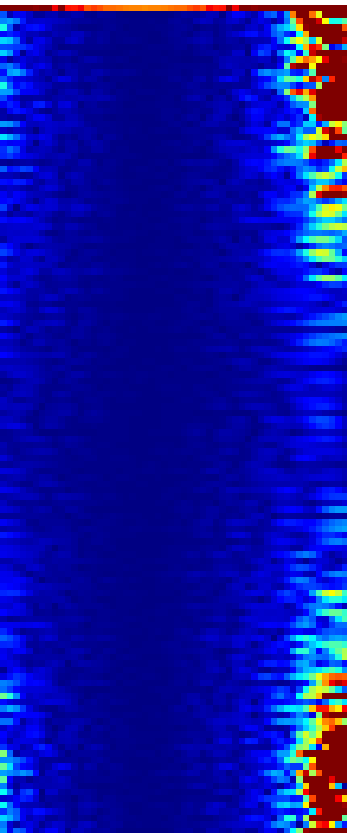
k_r

$C=10, t=1000$



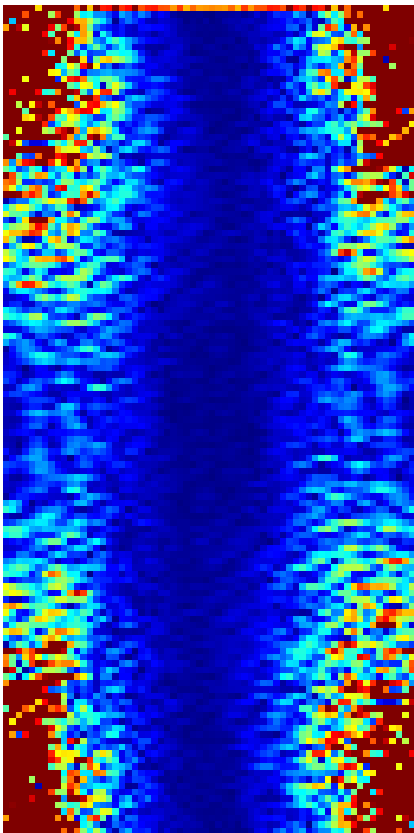
k_r

$C=1.5, t=750$



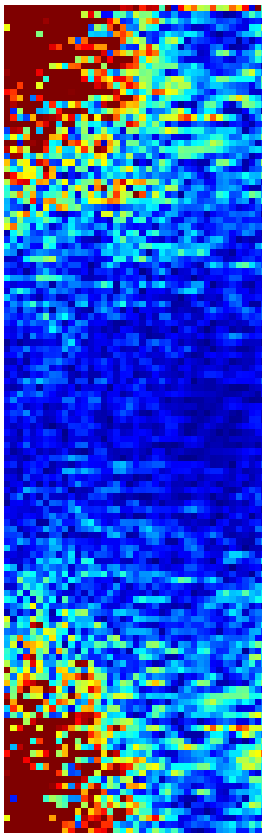
r

$C=1.5, t=1000$



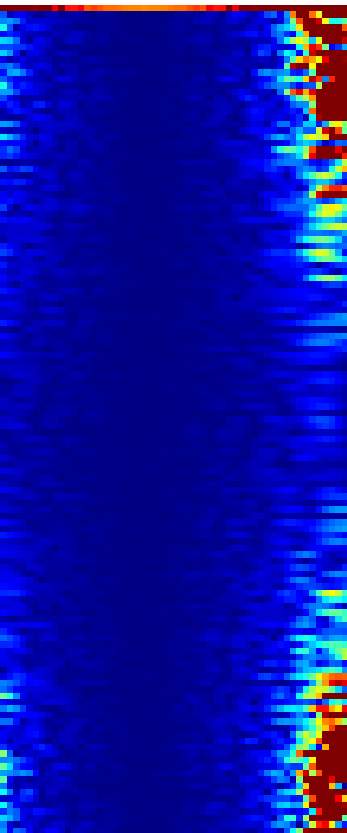
k_r

$C=1.5, t=1000$



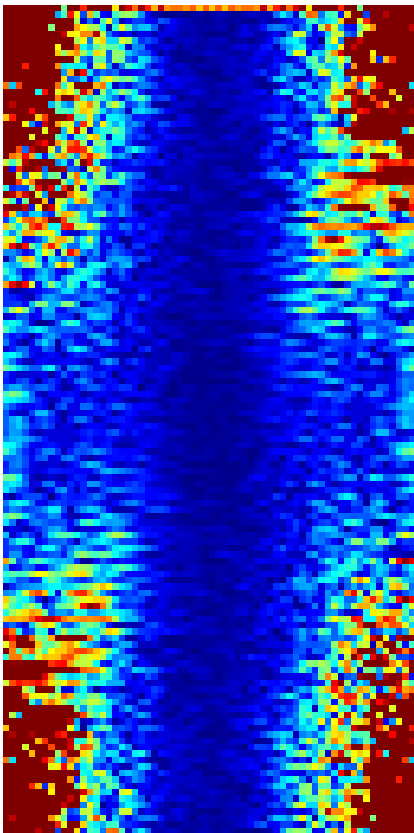
k_r

$C=2.5, t=750$



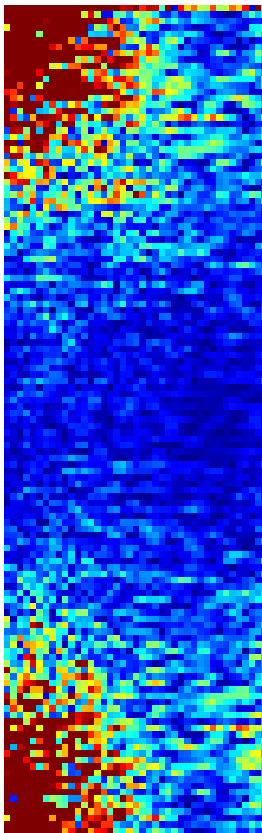
r

$C=2.5, t=1000$



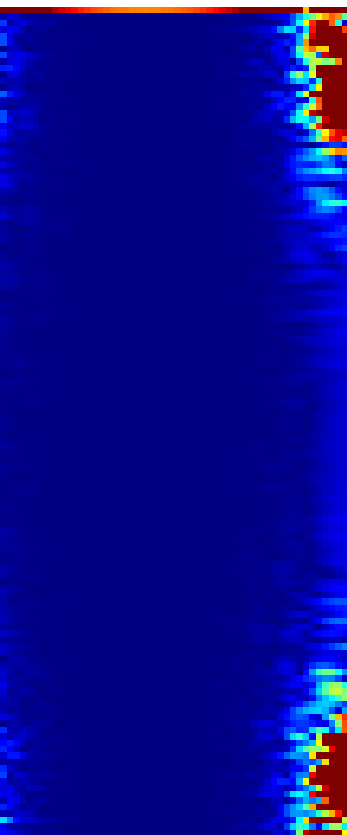
k_r

$C=2.5, t=1000$



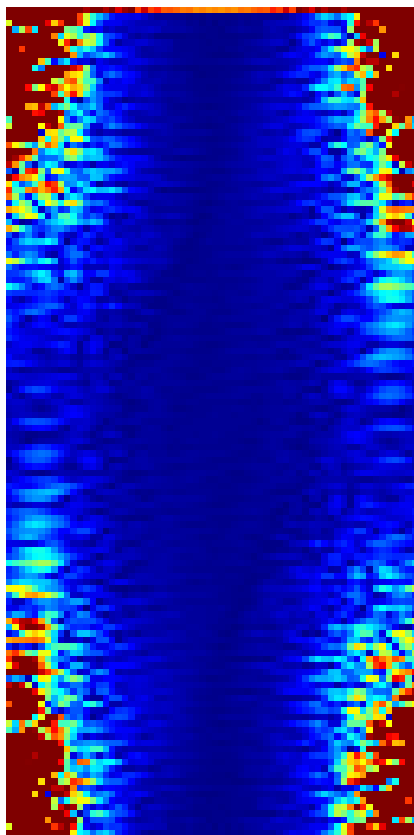
k_r

LAG,t=750



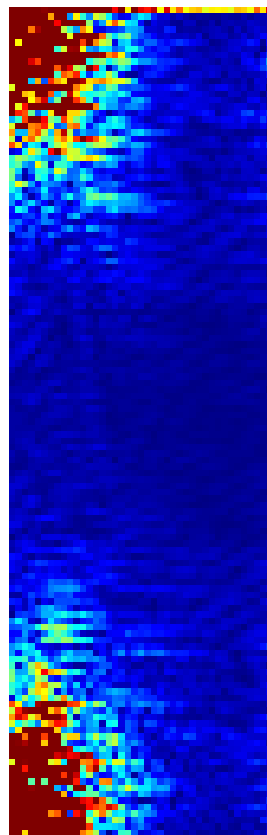
r

LAG,t=1000



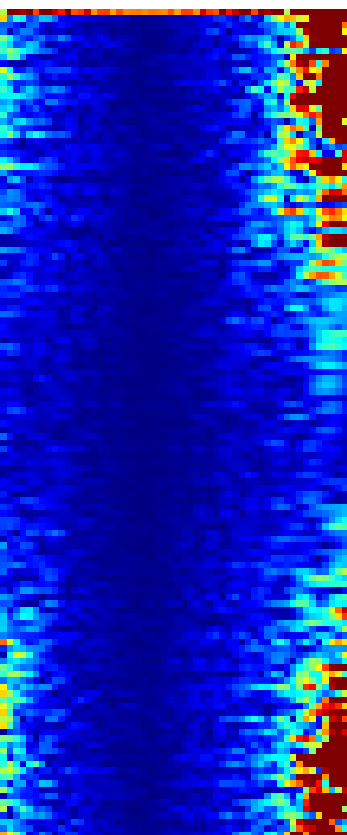
k_r

LAG,t=1500



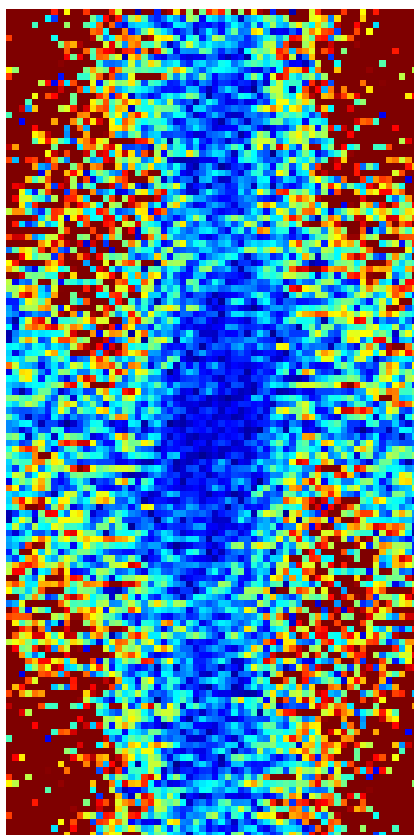
k_r

PSM,t=750



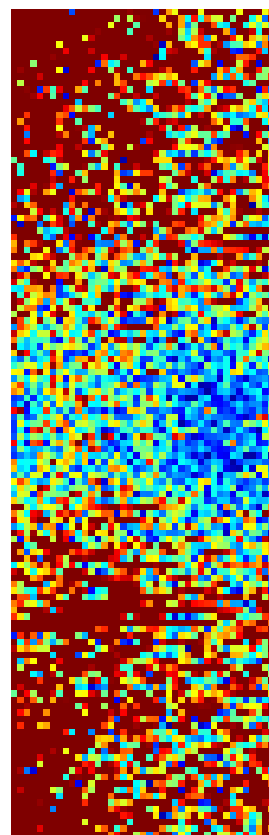
r

PSM,t=1000



k_r

PSM,t=1500



k_r